· 第2版 ·



Electronic Publication

指南 —— 从入门到放弃

系统了解ePub电子书制作

本入门教程介绍了ePub电子书在制作的过程中，
所需要掌握的知识、工具以及一些图文的排版技巧和规范。

赤霓 编著

The EPUB Makers of Alliance

VISIT...

LANZAROTE Caliente.com

· 第2版 ·



Electronic Publication

指南 —— 从入门到放弃

系统了解ePub电子书制作

本入门教程介绍了ePub电子书在制作的过程中，
所需要掌握的知识、工具以及一些图文的排版技巧和规范。

赤霓 编著

The EPUB Makers of Alliance

版权声明

本书的内容是作者多年制作ePub电子书的经验，在写书之际，引用了网络上公开的资料文档。本着互联网共享精神，现免费公之于网络，请不要把本书用于商业，谢谢！

关于本书的更新版本可发送邮件至1620279970@qq.com索要或添加QQ获取。

为获得最佳阅读效果，请在多看阅读设置中将排版设为「原版」（多看2.x版本）或「无」（多看3.x版本以上），背景为预设背景（不要自定义背景和字体颜色，以免整体配色出问题）；字体设置为「默认」（使用书中指定字体），字体大小为默认大小（一般手机上为＋3，平板上为＋2——即减小字体到最小值后，点击增大按钮的次数）。

前言

目前网络上流传的电子书，大部分只是粗略的进行了排版，基本谈不上良好的阅读体验，而精美排版的电子书还比较少。于是借此机会在总结经验的同时，采纳我的朋友　@笔杆　的建议，尽量扩展内容，让后学者少走一些弯路。

本书第1版从2017年2月13日开始写，于6月12日更新到最后一版——1.14。第1版完成以后，又经过14次的修改，内容逐渐臃肿，逻辑结构也有点混乱。于是重新梳理大纲，在7月25日开始进行修订，也就是第2版，最终在国庆假期完成。

面向读者

本书适合以下人群阅读：

- 喜欢用电子设备看书，但是不满意当前的排版。
- 从事电子书制作的同学。
- 希望了解电子书排版的数字出版工作者及开发者。
- 有一定制作基础，想要学习更多相关技术的人员。

对于电子书制作人员来说，最重要的一点是真心喜欢书籍并可以忍受枯燥重复的操作。如果你具备这一点，稍微用心地学习下，就能很快掌握制作电子书的技巧。

如何阅读本书

第2版与第1版的内容结构上一样，分为三大块：知识、技能、工具，细分之后又有12个章节。分别涉及制作ePub电子书的方方面面，可算作我对制作ePub的一次全面总结。本书一共分为正文12章、附录5章、参考资料1章。

其中1～4章主要讲解在制作电子书中需要学习掌握的知识：「第1章　ePub概述」需要了解一本电子书的主要构造；「第2章　HTML知识」和「第3章　CSS知识」是前端基础，要求掌握常用的标签和属性；「第4章　正则表达式」是比较重要的一章，学会之后，可以大大提升制作的效率。

5～8章主要讲解在制作电子书中可能会用到的编辑软件：「第5章　ePub编辑——Sigil」是必须要认真学习的一款软件，我们90%的操作都在这个软件中完成，推荐没有接触过ePub的电子书爱好者从这章开始学习；「第6章　图像处理——Photoshop」一章只需要用到Photoshop的比较少的几个功能；「第7章　文本处理——EmEditor」一章主要讲制作前的文本处理，配合正则表达式可以提升制作的效率；「第8章　制作电子书的辅助工具」一章，重点讲的是Sigil的插件功能和ePub电子书转PDF。

9～10两章：学习「第9章　排版知识」一章，可以从传统的出版中吸取营养，毕竟ePub是新兴的出版流程；「第10章　排版样例」一章介绍常见的内容排版技巧，并提供可以直接套用的样式。

「第11章　SMIL电子书（听书）」一章介绍的是一种新的出版物，优势在于可以同步发声，而不是简单的配上音频。这对于儿童电子读物、语言教学学科（如语文、英语）类的电子书，是非常重要的。

第12章主要是讲解不同格式的资源如何加工成ePub电子书，里面所讲的内容均是基于我可以接触到的工具。更多专业的输出工具，因为无法接触到，不了解其工作方式。也许它们的效率，会比我写出来的教程更加快捷方便。

附录5章是我在网络上搜集的资料，对于想要学习制作ePub的同学，有一定的参考价值。

参考资料是我在学习制作ePub的过程中，找到的零碎的教程，如果想要了解，也可以去看一看。

最后的话

如果这个入门指南，能带着你制作出一本个人满意的作品，我也会非常的开心！

另外，关于书名《ePub指南——从入门到放弃》，来自于我的朋友@王萌萌 的一句话。所谓的放弃，是不要忘记做书的目的是更舒适的阅读，当你领悟了之后，就可以放弃研究如何更精益求精的做书。因为一些简单规范的排版，就可以达到比较好的阅读体验。不要像我一样，沦为一个做书匠，做书如山倒，看书如抽丝。

赤霓
2017年10月3日

致谢



本书在还是第1版时，是

@笔杆 建议我扩充内容，让更多人受益；

@zecy 录制的Sigil视频教程，让我受益匪浅；

@Dave Heiland 在翻译Sigil英文版指南时，加深了对Sigil的理解；

@云轩阁主 在很早的时候就开始制作适配多看阅读器的电子书，我在他的博客中学习了很多排版技巧；

@子瑜 感谢他帮我校正本书。

还有一些人，在网络上留下的文章、资料对我的帮助也很大，具体可到「参考资料」中查看。

排版说明

书中存在实例代码、代码演示效果、排版样例等多种排版效果，这里进行说明下。

代码案例

代码案例配色方案取自Sublime Text默认主题Monokai，字体为Source Code Pro-Regular。显示效果如下：

```
<spine toc="ncx">
  <itemref idref="Chapter1.xhtml"/>
  <itemref idref="Chapter1-1.xhtml"/>
  ...
  <itemref idref="nav.xhtml" linear="no"/>
</spine>
```

测试效果

测试效果是代码案例的运行效果，以下面样式标出：

这是一个简单的正文段落。

段落元素由p元素定义。

排版样例

排版样例是指可以直接运用到电子书中的样式，提供样式的预览：

安徽籍工部左侍郎吕贤基上奏说：

今日事势，譬之于病，元气血脉，枯竭已甚，外邪又炽，若再讳疾忌医，愈难为救。为今之计，不妨效仿祖宗办法，着令各省官绅自办团练，则国体可报，匪患亦不难消矣。

此疏一上，咸丰帝茅塞顿开，马上采纳。大清国于是遍地办起团练。

咸丰三年（公元1853年）二月，太平军攻破安徽省城安庆，巡抚蒋文庆战殁，帮办团练大臣周天爵败走。太平军一鼓作气连占含山、巢县、和州之裕溪口。

行内代码

行内代码样式参考了 @廖雪峰 编程教程网站的排版，显示效果如下：

`font-size: 16px;`，字号。单位px，像素。默认16px。

超链接

本书中超链接颜色显示效果如下：

导航文档的结构和语义定义见ePub导航文件。

Tips

本书中需要注意或者提示的内容，样式如下：

> `unique-identifier`的值必须和`dc:identifier`元素的ID属性匹配。这一点在10.11.4节有体现。

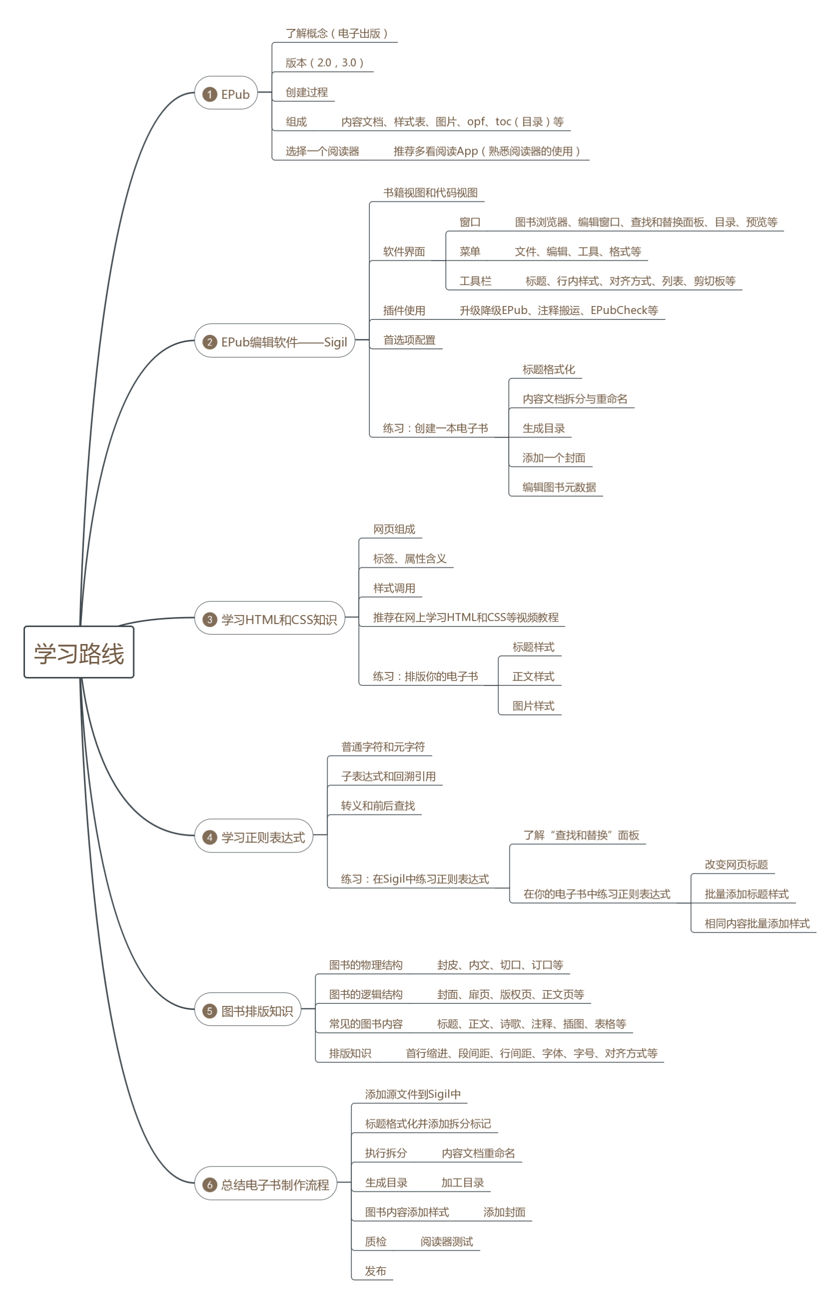

学习路线
1 EPub
了解概念（电子出版）
版本（2.0，3.0）
创建过程
组成
内容文档、样式表、图片、opf、toc（目录）等
选择一个阅读器
推荐多看阅读App（熟悉阅读器的使用）
2 EPub编辑软件——Sigil
书籍视图和代码视图
软件界面
窗口
图书浏览器、编辑窗口、查找和替换面板、目录、预览等
菜单
文件、编辑、工具、格式等
工具栏
标题、行内样式、对齐方式、列表、剪切板等
插件使用
升级降级EPub、注释搬运、EPubCheck等
首选项配置
练习：创建一本电子书
标题格式化
内容文档拆分与重命名
生成目录
添加一个封面
编辑图书元数据
3 学习HTML和CSS知识
网页组成
标签、属性含义
样式调用
推荐在网上学习HTML和CSS等视频教程
练习：排版你的电子书
标题样式
正文样式
图片样式
4 学习正则表达式
普通字符和元字符
子表达式和回溯引用
转义和前后查找
练习：在Sigil中练习正则表达式
了解“查找和替换”面板
在你的电子书中练习正则表达式
改变网页标题
批量添加标题样式
相同内容批量添加样式
5 图书排版知识
图书的物理结构
封皮、内文、切口、订口等
图书的逻辑结构
封面、扉页、版权页、正文页等
常见的图书内容
标题、正文、诗歌、注释、插图、表格等
排版知识
首行缩进、段间距、行间距、字体、字号、对齐方式等
6 总结电子书制作流程
添加源文件到Sigil中
标题格式化并添加拆分标记
执行拆分
内容文档重命名
生成目录
加工目录
图书内容添加样式
添加封面
质检
阅读器测试
发布

Table of Contents

版权声明
前言
致谢
排版说明
学习路线
目录
第1章　ePub概述
　1.1　什么是ePub？
　　1.1.1　创建过程
　　1.1.2　使用技术
　　1.1.3　规范文档
　1.2　组成ePub的文件
　　1.2.1　mimetype和container.xml
　　1.2.2　content.opf
　　1.2.3　toc.ncx
　　1.2.4　核心媒体类型
　1.3　选择一个阅读器
　1.4　几种常见的电子书格式
　1.5　无障碍访问
　　1.5.1　导航
　　1.5.2　语义标记
　　1.5.3　动态布局
　　1.5.4　有声播报和媒体叠加
　　1.5.5　备选方案
　　1.5.6　脚本
第2章　HTML知识
　2.1　块状元素
　　2.1.1　p（段落）
　　2.1.2　div（分区或节）
　　2.1.3　h1~h6（标题）
　　2.1.4　table（表格）
　　2.1.5　ol、ul（列表）
　　2.1.6　blockquote（引用）
　　2.1.7　section（节）
　　2.1.8　figure（流内容）
　　2.1.9　aside（内容之外的内容）

2.1.10 pre（预格式）
2.1.11 audio（音频）
2.1.12 video（视频）
2.1.13 hr（水平线）
2.1.14 nav（导航）
2.2 内联元素
2.2.1 br（换行）
2.2.2 span（行内组合）
2.2.3 b（粗体）
2.2.4 i（斜体）
2.2.5 u（下划线）
2.2.6 del（被删除）
2.2.7 sub（下标）
2.2.8 sup（上标）
2.2.9 a（超链接）
2.2.10 img（图像）
2.2.11 ruby（行间注）
2.3 其他元素
2.3.1 html（网页文档）
2.3.2 head（头部）
2.3.3 title（文档标题）
2.3.4 link（外部资源）
2.3.5 style（样式）
2.3.6 body（主体）
2.4 字符实体
第3章 CSS知识
3.1 样式来源
3.2 层叠规则
3.3 选择器
3.3.1 通用选择
3.3.2 单标签
3.3.3 多标签
3.3.4 id选择器
3.3.5 类选择器
3.3.6 属性选择器
3.3.7 伪类
3.3.8 伪元素
3.4 属性和值
3.4.1 动画属性
3.4.2 背景属性
3.4.3 边框属性
3.4.4 opacity属性
3.4.5 尺寸属性
3.4.6 字体属性
3.4.7 列表属性
3.4.8 外边距属性

3.4.9 内边距属性
3.4.10 多列属性
3.4.11 定位属性
3.4.12 打印属性
3.4.13 表格属性
3.4.14 文本属性
3.5 尺寸
3.6 书写规范
第4章 正则表达式
4.1 字符
4.1.1 普通字符
4.1.2 特殊元字符
4.1.3 基本元字符
4.1.4 数量元字符
4.1.5 位置元字符
4.1.6 回溯引用和前后查找
4.2 转义与回溯引用
4.3 字符组
4.4 应用
4.5 阅读正则表达式
第5章 ePub编辑——Sigil
5.1 介绍——Sigil：ePub编辑器
5.1.1 更新
5.1.2 关于Sigil
5.1.3 关于ePub
5.2 安装——入门
5.3 功能——深入指南
5.3.1 用户界面：窗口、菜单和工具栏
5.3.2 首选项：配置Sigil
5.3.3 打开并保存文件：加载并保存您的书籍
5.3.4 书籍视图：所见即所得的编辑
5.3.5 代码视图：直接编辑HTML
5.3.6 预览：预览代码视图的修改
5.3.7 图书浏览器：管理ePub中的文件
5.3.8 元数据：编辑书籍的详细信息
5.3.9 添加封面：为您的书选一张封面
5.3.10 目录：一本书的地图
5.3.11 验证：确保您的ePub是无错误的
5.3.12 拼写检查：检查和纠正拼写
5.3.13 图像、视频、音频：将媒体文件插入到书中
5.3.14 特殊字符：插入特殊字符
5.3.15 拆分与合并：操作章节和文件
5.3.16「查找&替换」：搜索代码视图
5.3.17 搜索模板：保存的「查找&替换」搜索
5.3.18 剪切板：保存文本和样式
5.3.19 剪贴板历史：剪贴板列表

5.3.20 链接和ID：链接、注释和ID
5.3.21 样式：使用样式表
5.3.22 索引：为您的书创建一个索引
5.3.23 报告：查看有关您书的详细信息
5.3.24 外部编辑：使用「打开方式」编辑文件
5.4 教程——如何利用该指南
5.4.1 ePub入门
5.4.2 准备在Sigil中打开的文件
5.4.3 用Sigil打开你的文件
5.4.4 保存您的ePub文件
5.4.5 添加作者和标题
5.4.6 添加封面图片
5.4.7 为每一章创建单独的文件
5.4.8 识别你的章节
5.4.9 生成目录
5.4.10 创建链接和注释：链接您的文本
5.4.11 快速介绍代码视图
5.4.12 检查拼写错误
5.4.13 检查ePub中的错误
5.4.14 使用「查找&替换」
5.4.15 使用样式表：使用样式格式化
5.4.16 自定义字体：在书中嵌入字体
5.4.17 高级「查找&替换」——正则表达式
5.4.18 正则表达式参考
5.4.19 提示
5.5 常问问题
5.5.1 常见问题
5.5.2 在哪里获得帮助
5.5.3 使用Sigil
5.5.4 格式和样式
5.5.5 如何删除段落之间的空格？
5.6 为Sigil做贡献：帮助发展
第6章 图像处理——Photoshop
6.1 图片知识
6.1.1 格式
6.1.2 模式
6.2 使用「动作」处理图片白边
6.3 一张可用的封面
6.3.1 图书馆中获取封面
6.3.2 立体封面修改为平面图片
6.3.3 封面重制
6.3.4 修改封面尺寸
6.4 生僻字图片制作
第7章 文本处理——EmEditor
7.1 常规设置
7.2 配置Text

7.3　批处理
7.4　批处理应用（配合多看读校文本）
7.5　宏脚本
7.6　大纲栏
第8章　制作电子书的辅助工具
8.1　Adobe Acrobat——编辑PDF
8.1.1　添加书签
8.1.2　文字识别（OCR）
8.2　pagina-EPUB-Checker——验证ePub
8.3　Fontmin——字体子集化方案一
8.4　FontRepack——字体子集化方案二
8.5　subAllFont——字体子集化方案三
8.6　Sigil的常用插件
8.6.1　DOCX文档转ePub
8.6.2　ePub转Mobi
8.6.3　ePub2与ePub3互转
8.6.4　使用EpubCheck插件验证ePub文件
8.6.5　搬运注释
8.6.6　段落中注释修改成跳转注释
8.6.7　Sigil插件清单
8.7　Prince——ePub转PDF
8.7.1　页面布局
8.7.2　分页媒体
8.7.3　页面大小
8.7.4　裁剪标记
8.7.5　页面区域
8.7.6　打印颜色
8.7.7　色彩管理
8.7.8　打印图像
8.7.9　生成内容
8.7.10　高级分页媒体
8.7.11　PDF书签
8.7.12　ePub转PDF演示
8.8　Web Equation——ePub中插入数学公式
第9章　排版知识
9.1　Web排版
9.2　平面设计4原则
9.3　术语
9.4　字体
9.5　字号
9.6　标点符号
9.7　符号
9.8　装饰符号
9.9　开本
9.10　一般图书内容的顺序
9.11　字符编码

9.12 中文字体英文名称
9.13 颜色的选择与搭配
9.13.1 颜色的种类
9.13.2 颜色的印象
9.13.3 底色与文字颜色
第10章 排版样例
10.1 目录格式
10.2 标题
10.3 正文
10.3.1 一般正文
10.3.2 书信
10.3.3 诗词曲
10.3.4 古体诗歌自适应换行
10.4 插图
10.4.1 单张插图
10.4.2 交互图
10.4.3 并列图
10.4.4 画廊图
10.4.5 行内图
10.4.6 图片绕排
10.4.7 透明图昼夜自适应
10.5 封面
10.5.1 多看阅读
10.5.2 掌阅
10.5.3 Kindle
10.6 注释
10.6.1 注释写在alt或扩展属性中
10.6.2 超链接注释
10.6.3 多看阅读弹注
10.6.4 ePub3标准注释
10.6.5 汉字注音
10.7 多媒体
10.7.1 音频
10.7.2 视频
10.8 表格
10.9 竖排
10.9.1 支持ePub3的阅读器
10.9.2 多看阅读
10.9.3 浮动方式实现竖排
10.9.4 Kindle的竖排问题
10.10 生僻字
10.10.1 替换
10.10.2 偏旁组合
10.10.3 图片
10.10.4 外挂字库
10.10.5 自造字

10.10.6　图片字做字体
10.11　多看扩展效果
10.11.1　贴边
10.11.2　全屏展开
10.11.3　波浪下划线
10.11.4　正文字体自定义
10.11.5　使用多看字体
10.11.6　幻灯片（漫画）
10.11.7　多看书城漫画（旧）
第11章　SMIL电子书（听书）
11.1　SMIL文件语法
11.2　创建流程
11.3　使用Audacity标记时间
第12章　不同格式的文档制作ePub
12.1　排版文件与版式文件
12.2　TXT文本制作ePub
12.2.1　文档整理
12.2.2　内容整理
12.2.3　拆分正文
12.2.4　生成目录
12.2.5　封面和版权
12.2.6　辅文修改
12.2.7　正文部分
12.2.8　注释内容
12.2.9　最后工作
12.3　Word文档制作ePub
12.3.1　整理代码
12.3.2　生成目录
12.3.3　辅文排版
12.3.4　正文排版
12.3.5　制作总结
12.4　图像PDF转TXT格式文本
14.1　图片处理
14.2　开始OCR图书
14.3　处理文本错误
14.4　比对文本
14.5　合并段落
12.5　AZW3/MOBI制作ePub
12.5.1　Kindle电子书转ePub
12.5.2　整理文件及确定目录
12.5.3　删除无用的代码
12.5.4　精加工
12.5.5　最后工作
12.6　常见问题
12.6.1　网络小说的制作
12.6.2　合并多个ePub电子书

附录一　epub:type属性定义
附录二　多看阅读的教案
- 图片规范
  - 全屏图
  - 非全屏图
  - 图片与文件命名时所用符号规则
  - 路径
- 用户自建CSS样式说明
  - 样式类型
  - CSS结构
  - 各种样式类型内部的样式命令
- 元数据编辑器相关说明
- 默认样式表

附录三　自用字体样式表
附录四　掌阅的排版规范
- 质检标准
- 默认样式表

附录五　ePub3的案例和正规教程
附录六　成品要求（推荐标准）
参考资料
赞助
版本记录

Lists of Table

表1　META-INF 文件夹的特殊配置文件
表2　移动端ePub阅读器推荐
表3　HTML的字符实体
表4　正则表达式普通字符
表5　正则表达式特殊元字符
表6　正则表达式基本元字符
表7　正则表达式数量元字符
表8　正则表达式位置元字符
表9　回溯引用和前后查找
表10　Sigil中内容文档命名规则
表11　页面大小关键字及尺寸
表12　页面区域的名称及默认属性
表13　Web与数字出版的区别、联系
表14　字号、pt、px、em之间的换算
表15　标点符号的名称和术语
表16　常用数学符号
表17　拉丁文字母表
表18　德文字母表
表19　俄文字母表
表20　希腊文字母表
表21　常用术语简写词表
表22　常用术语简写词表
表23　书刊成书和版心尺寸
表24　Windows与OS X系统的默认字体
表25　Kindle的内置字体名称
表26　竖排与横排之比较
表27　竖排与横排标点的使用
表28　多看阅读器字体CSS写法
表29　网文常见标题格式
表30　样式的组成
表31　span元素支持的属性（多看）
表32　div元素支持的属性（多看）
表33　img元素支持的属性（多看）
表34　ol元素支持的属性（多看）
表35　ul元素支持的属性（多看）
表36　table元素支持的属性（多看）

表37　td元素支持的属性（多看）
表38　a元素支持的属性（多看）

第1章

ePub概述

目前市面上的电子书大多是ePub格式的，随着ePub格式的推广，会有越来越多的人因为不满足现有电子书的排版，而加入学习制作ePub电子书来。本章内容介绍ePub的基础性知识，来帮助不了解ePub电子书的同学对ePub格式有一个粗略的印象。

ePub格式电子书是被广泛采用的电子图书格式，最新的规范增强了ePub格式电子书的功能，以更好地支持多样性出版物的要求，包括复杂的布局、富媒体互动、全局排版功能。ePub 3可以应用于更加广泛的内容：图书、杂志、教育专业和科学出版物。

1.1 什么是ePub？

ePub是由国际数字出版论坛（International Digital Publishing Forum，简称为IDPF）提出的一种自由开放的电子图书标准，使用它可以呈现固定布局排版的文档，也可以实现内容自动重排。内容重排的最大优势在于可以根据阅读设备的尺寸用最适于用户阅读的方式呈现作品。

ePub格式是在1999年开放电子图书论坛（Open eBook Forum，简称OeBF，后来OeBF改名为「IDPF」）提出的OEBPS格式基础上，经过不断研究和改进而形成的。1999年9月，OeBF正式发布开放电子图书出版结构规范1.0版（Open eBook Publication Structure 1.0，简称OEBPS 1.0）。

随后，OEBPS1.0经历了两次更新。一次是2001年6月，OeBF出版结构工作组发布OEBPS 1.0.1；另一次是，2002年8月发布了OEBPS 1.2。2003年IDPF着手制定OEBPS 2.0规范，但因种种原因未能实现目标，这项工作后来成为EPUB2.0标准的基础。在前期工作的基础上，为了对OEBPS制定一个单一的文档容器格式，2006年10月IDPF发布OCF 1.0（Open Container Format），OCF 1.0也可以说是最初的EPUB 1.0。2007年9月IDPF发布OPF 2.0（Open Packaging Format）及OPS 2.0（Open Publiccation Structure），也即我们说的EPUB 2.0标准。

2009年，IDPF开始组织人力修订EPUB标准，在2010年7月正式发布EPUB 2.0.1。EPUB 2.0.1由OPS 2.0.1、OPF2.0.1和OCF2.0.1组成。2011年2月15日，IDPF发布了历时7个月完成的EPUB 3.0第一草案，随后5月23日发布EPUB 3.0格式规范建议细则，并开始接受公众评议。2011年10月IDPF会员聚集在法兰克福开会，投票通过了最终版本的EPUB 3.0格式标准细则。

2014年6月26日，EPUB 3.0.1发布，作为3.0的小维护更新。2017年1月5日发布EPUB 3.1，对文字的格式规范进行了重组和清理。

2017年1月30日，IDPF与W3C合并（新闻稿），作为会员组织存在的

IDPF已停止运营。该网站只作为一个档案保存站点，有关EPUB的更多信息请访问PUBLISHING@W3C。



图1-1　IDPF标志

1.1.1　创建过程

快速了解ePub的方法是分解其制作的过程：

1. 制作ePub的第一步是创建内容文档，可以是XHTML、SVG或两者的混合。如果您之前就创建了网页，再制作ePub会比较简单。如果没有，有许多编辑器可以创建XHTML文件，也有一些程序可以生成一个完整的ePub。但是请注意，内容文档通常会引用许多其他类型的资源，如CSS样式表、图像、视听文件、字体等，用于补充或渲染网页，所有这些文件需要分类放置在不同的文件夹中，见1.2节。
2. 内容文档制作完成之后，下一步是创建包文件（*content.opf*），这是阅读系统用于读取有关出版物信息的特殊文档。创建此文件的第一步是在包文件中列出内容文档中使用的所有资源清单（大部分资源将储存ePub文件中，但有些如音频和视频，可以在线调用）。阅读系统需要此列表来确定资源是否完整，并检索发现远程调用的文件。所有出版物元数据（标题、作者等）在一个单一的元素中（`<metadata>`），以便在阅读系统中轻松提取和使用。您还必须在书干部分（`<spine>`）中调整默认阅读顺序——内容文件的顺序列表，从第一个到最后一个依次排序。最后，包文件确定哪个文档资源代表导航文件（目录）。
3. 最后一步是将您的内容文档、其他类型的资源和包文档压缩成一个为**.epub*扩展名的文件。这个过程并不像普通压缩那样：必须先添加一个特殊的mimetype文件，以表示zip文件为一个ePub，而不是别的东西；一个在名为META-INF目录文件夹中的*container.xml*的文件，用于告诉阅读系统在哪里找到您的包文件。

以上就是手动制作一个ePub电子书的过程。手工制作ePub的教程见此文章——使用EPUB制作数字图书。

阅读系统的工作原理：检查您的zip文件，确定它是一个ePub，找到包文件，并从那里找到如何将资源呈现给读者的方式。

1.1.2　使用技术

ePub中使用的一些技术是由国际数字出版论坛专门开发的，该组织是维护ePub规范的组织，其成员包括各种出版发行商和阅读系统的开发人员。但是，大多数被采用的标准都得到国际认可，至少是大多数读者熟悉的。您将在ePub 3出版物中找到的关键技术包括：

- HTML5：用于表示文本和多媒体内容，现在包括对MathML方程式的本地支持，ruby发

音标记和嵌入式SVG图像

- SVG：表示可缩放矢量图形（例如漫画）
- CSS：显示和呈现视觉内容
- JavaScript：用于交互和自动化
- TrueType和WOFF：提供超出阅读系统的字体支持
- SSML/PLS/CSS 3 Speech：文本到语音的呈现
- SMIL：用于音频播放时同步文本
- RDFvocabularies：用于嵌入关于出版物和内容的语义信息
- XML：一个专门的语法，方便编辑处理ePub
- ZIP：将所有资源整合到单个文件中

在制作ePub中，我们使用的最多的是XHTML和CSS，这两个是重点需要学习的。其他如JS、字体、SVG、XML等只需要明白会用即可。

1.1.3　规范文档

ePub 3.1的核心标准主要有六个规范组成，它们分别是：

1. ePub包规范（ePub Packages 3.1）。该规范对书卷所包含的内容以及与出版物资源相关联的设置进行描述，主要包括了包的元数据、包文档和导航文档内容的规范。
2. ePub内容文档规范（ePub Content Documents 3.1）。该规范定义了出版物中使用的XHTML、SVG、CSS文件以及其中含有的脚本、固定布局、词汇发音的标准。
3. ePub媒体叠加规范（Media Overlays 3.1）。本规范通过SMIL（Synchronized Multimedia Integration Language）同步多媒体集成语言定义了文本和音频同步的格式和处理模型，对ePub出版物中媒体层定义及媒体的建立和播放行为等内容的描述，用于要求文字与媒体同步的数字出版物。
4. ePub开放容器格式规范（Open Container Format 3.1），该规范简称OCF，是ePub标准中一个非常重要的容器技术规范，它定义了ePub出版物的文件格式和处理模型，并将出版物所有相关内容按规范打包封装在单一的文档容器中，同时还定义了一种出版物资源混淆嵌入的方法。本规范中定义的容器包括抽象容器和ZIP容器两部分。OCF抽象容器是一种文件系统模型，它要求ePub出版物内所有内容使用一个共同的根目录中，所有本地资源都位于根目录所在的目录树中，但不强制资源目录的结构形式。抽象容器模型要求在根目录下强制包含一个名为META-INF文件夹，用于存储特殊的配置文件，相关配置文件的名称和作用见表1。抽象容器模型在根目录中的强制包含一个文件名「mimetype」的文件，保留供OCF ZIP容器使用。抽象容器模型对资源文件的要求较为宽松，理论上OCF抽象容器中所有资源文件可以位于根目录下的任何位置，只要它们不在META-INF文件夹中即可，但是为方便对出版物的解析，建议将EPUB出版物的内容存储在根目录下的一个专用文件夹中。OCF ZIP容器规定了最终的出版物采用一个ZIP压缩的文件来进行发布，文件扩展名为*.epub*。OCF允许使用「Deflate」算法或「不压缩」两种情况进行文件压缩，之外的算压不支持，不能进行加密压缩，不允许分卷压缩。
5. ePub可访问性规范（ePub　Accessibility）。该规范定义为了出版物定义了可访问性、一致性和发现元数据（Discovery Metadata）的标准。本规范不针对单个版本的EPUB。它旨在适用于符合任何版本或配置文件（包括未来版本的标准）的ePub出版物。
6. 替代样式标签规范（Alternate Style Tags）。定义了使用微格式标记替代样式表的模式。注

表1　META-INF 文件夹的特殊配置文件

文件名	必要性	作用
container.xml	必须	标识定义 *.opf包文档。
signatures.xml	可选	包含资源的数字签名。
encryption.xml	可选	包含资源加密或混淆相关的信息。
metadata.xml	可选	用于存储OCF ZIP容器的元数据。
rights.xml	可选	用于存储数字版权管理信息。
manifest.xml	可选	资源内容的列表清单。

ePub 3.1的可选规范主要有：ePub规范片段标识符注（Canonical Fragment Identifiers）、ePub词典和词汇表（Dictionaries and Glossaries）、ePub索引（ePub Indexes）、ePub多出版物呈现（Multiple Renditions）、ePub预览（Previews）和ePub区域导航（Region-Based Navigation）等，用于增强ePub出版物的功能和用户体验。

EPUB 3.1标准见IDPF网站，比较旧的中文版翻译见ePub 3.0 Standard Chinese Version网站，资料由AnFengDe网站提供，感谢。

1.2　组成ePub的文件

ePub®规范的最大特点是基于Web标准，它实现了将包括XHTML、CSS、SVG、图像文件及其他媒体资源在内的多种格式的内容结构化地压缩打包为一个单一文件，便于数字出版物的分发与交换。

出版物资源通常压缩为*.zip*存档来发布，文件扩展名为*.epub*。作为合法的zip压缩文件，出版物可以被许多程序解压，因此它们的制作和编辑得以简化。为了直观地了解一个*.epub*文件的结构，我解压了本书，它的文件结构（部分）是：

```
ePub入门.epub
│   mimetype
├─META-INF
│      container.xml
└─OEBPS
     │ content.opf
     │ toc.ncx
     ├─Audio
     │      Tiger.mp3
     ├─Fonts
     │      HuangCao.ttf
     ├─Images
     │      001.png
     │      002.png
     │      003.png
     │      timg.jpg
     ├─Styles
     │      Stylesheet.css
     ├─Text
     │      Cover.xhtml
     │      Chapter1.xhtml
     │      Chapter2.xhtml
     │      Chapter3.xhtml
     └─Video
             sunup.mp4
```

下面章节分别介绍各类型文件含义及用途。

1.2.1 mimetype和container.xml

这两个文件在ePub电子书中都是必须要有的文件，不可缺少。

mimetype

它的内容较为简单，仅对压缩包的打开方式进行了说明，其文件内容为「application/epub+zip」，表示出版物的压缩包文档可以用epub和zip两种方式来进行解析使用。实事上ePub格式的出版物将扩展名由*.epub*改为*.zip*后，可以使用相应的解压缩软件解压或打开，并且不影响文件内容。此文件在Sigil中创建一本书时，软件会自动生成。

container.xml

ePub阅读系统首先解析和查看该文件，<rootfile>标签代码里指明了包文件的文件名、路径和媒体类型，其中full-path="..."指定了包文档的文件名和路径，可根据实际修改。默认包文件名为*content.opf*。

```
<?xml version="1.0" encoding="UTF-8"?>
<container version="1.0" xmlns="urn:oasis:names:tc:opendocument:xml
  <rootfiles>
    <rootfile full-path="OEBPS/content.opf" media-
type="application/oebps-package+xml"/>
  </rootfiles>
</container>
```

*container.xml*文件存放在META-INF文件夹下。该文件同样由Sigil创建，制作者不需要做任何处理。

1.2.2 content.opf

每个ePub都有一个单一的包文件，它的扩展名必须为*.opf*，内部文件格式采用XML标准，通过包文件可以准确识别整个数字出版物的结构及阅读顺序，是解析出版物及显示ePub出版物的主要信息来源。包文件放置在名为「OEBPS」的文件夹下，导航文件及其他媒体文件同样存储在此文件夹中。

package标签

<package>是包文件的根标签，它封装出版物元数据和资源信息。

<package>标签有version、unique-identifier、dir、id、prefix和xml:lang等六个属性，其中version、unique-identifier是必须定义的。version属性值表明ePub文件所采用ePub规范的版本，unique-identifier属性值设

定了对文件内<dc:identifier>标签的引用，以获取作品的唯一标识符，如UUID、DOI、ISBN 或ISSN等。

<package>标签内按照顺序依次包含<metadata>、<manifest>、<spine>和<collection>等四个标签。其中<metadata>、<manifest>和<spine>三个为必备的标签。<metadata>标签提供出版物基本信息和内容布局的显示信息，便于检索书目信息和访问控制。

metadata标签

<metadata>标签是用来储存电子书元数据的，简要介绍以下几种属性：

- language：中文版Sigil默认使用的zh-CN（简体中文），可以根据图书的语言自己修改；
- title：电子书的标题，一般指书名；
- author：电子书的作者；
- date：事件日期，可以设置三个内容。creation（制作日期，制作该电子书的日期）、modification（软件会自己记录最新的修改日期）、publication（出版日期，纸质出版物的出版日期）；
- identifier：识别符，在Sigil中创建一个ePub文件时，会自动生成一个识别符——UUID；也可以再增加一个ISBN属性用来记录纸质书的国际标准书号，如果有的话；
- subject：主题词或关键字；
- description：电子书的内容介绍；
- contributor：贡献者或其它次要责任者；
- source：图书资源或素材的来源；
- rights：版权描述；

以上信息可以直接在Sigil的「元数据编辑器」中直接添加，相关内容见5.3.8节。

manifest标签

<manifest>标签提供出版物内容资源的列表，每一个资源由一个<item>标签来表示。在Sigil中，该信息由软件自动生成，不需要制作者再修改。需要说明的是，<manifest>标签里<item>的顺序不重要。

本书中的<manifest>信息（部分）：

```
<manifest>
   <item id="ncx" href="toc.ncx" media-type="application/x-dtbncx+xml"/>
   <item id="Chapter1.xhtml" href="Text/Chapter1.xhtml" media-type="application/xhtml+xml"/>
   <item id="sgc-nav.css" href="Styles/sgc-nav.css" media-type="text/css"/>
   <item id="nav.xhtml" href="Text/nav.xhtml" media-type="application/xhtml+xml" properties="nav"/>
```

```
  <item id="Chapter1-1.xhtml" href="Text/Chapter1-1.xhtml" media-type="application/xhtml+xml"/>
  <item id="Chapter1-2.xhtml" href="Text/Chapter1-2.xhtml" media-type="application/xhtml+xml"/>
  <item id="Chapter1-3.xhtml" href="Text/Chapter1-3.xhtml" media-type="application/xhtml+xml"/>
  <item id="Chapter1-4.xhtml" href="Text/Chapter1-4.xhtml" media-type="application/xhtml+xml"/>
  <item id="Stylesheet.css" href="Styles/Stylesheet.css" media-type="text/css"/>
  <item id="AdobeGaramondLT-Regular.ttf" href="Fonts/AdobeGaramondLT-Regular.ttf" media-type="application/x-font-ttf"/>
  <item id="SourceCodePro-Regular.ttf" href="Fonts/SourceCodePro-Regular.ttf" media-type="application/x-font-ttf"/>
  <item id="epub-logo.png" href="Images/epub-logo.gif" media-type="image/png"/>
</manifest>
```

id表示文件的ID号；href指文件的路径信息；media-type指文件的媒体类型。

spine标签

ePub的一个关键概念是人或者程序可按某种特定的顺序对构成出版物的多个资源进行完全导航和加以使用。

许多出版物都有明显的阅读顺序或遍历内容的逻辑进程。一本小说是高度顺序化文件的例子，它通常有开始、中间和结尾。但不是所有的出版物都这样有序：一本烹饪书或影集可能被认为是更像数据库的。然而，所有文件在顶级内容项都至少有一个逻辑顺序，不论是按照日期、主题、地点或一些其它标准（例如，烹饪书通常是按照菜谱类型排列）。

`<spine>`标签的toc属性内容必须与`<manifest>`标签列出来的`id="ncx"`一致。`<spine>`标签提供书出版物内容资源的顺序列表，通过`<itemref>`子标签的顺序定义默认阅读顺序。当前电子书的阅读顺序（部分）：

```
<spine toc="ncx">
  <itemref idref="Chapter1.xhtml"/>
  <itemref idref="Chapter1-1.xhtml"/>
  <itemref idref="Chapter1-2.xhtml"/>
  <itemref idref="Chapter1-3.xhtml"/>
  <itemref idref="Chapter1-4.xhtml"/>
  <itemref idref="nav.xhtml" linear="no"/>
</spine>
```

linear属性的含义是指定引用的内容是否为主要内容。属性的值必须yes

或no。默认值是yes。

<spine>标签中的linear属性表明该项是作为线性阅读顺序中的一项，还是和先后次序无关。有些阅读器可以将spine中linear="no"的项作为辅助选项处理，有些阅读器则选择忽略这个属性，比如多看阅读。例如在下边的实例中，支持辅助选项处理的阅读器会依次显示chapter1、chapter3，而chapter2只在目录中点击到（或者其他开启动作）之后才会显示。

```
<spine toc="ncx">
  <itemref idref="chapter1.xhtml"/>
  <itemref idref="chapter2.xhtml" linear="no"/>
  <itemref idref="chapter3.xhtml"/>
</spine>
```

使用Lithium阅读器浏览这个示例时，在阅读区域chapter1之后即是chapter3，chapter2被忽略了。只有在目录中才可以看到chapter2，如果书中有超链接是链接到chapter2的话，也可以显示chapter2的内容。

spine标签有page-progression-direction=""设置出版物内容流动的全局方向。可设置的值：

- ltr：从左到右
- rtl：从右到左
- default（默认值）

当要制作一本竖排电子书的时候，才需要设置该属性。如：

```
<spine toc="ncx" page-progression-direction="rtl">
  ...
</spine>
```

在多看阅读器中实现竖排，见10.9节。

guide标签

content.opf内容文件的最后一部分是<guide>标签，<guide>可以为ePub阅读系统提供语义信息。OPF2的<guide>标签被废弃了，代之以*nav.xhtml*的landmarks特性。为了向前兼容ePub　2阅读系统，制作者可以在包文件里包括<guide>标签。

在Sigil软件中，打开「图书浏览器」窗口，选择Text文件夹下的网页文件，根据网页的内容选择相应的语义，见5.3.7节。步骤：选择要添加语义的网页，然后点击鼠标右键，选择「添加语义」菜单项，从语义列表中选择相应的条目。增加语义的内容文档，会在guide标签中显示如下

（示例）：

```
<guide>
  <reference type="cover" title="封面" href="Text/Cover.xhtml"/>
  <reference type="foreword" title="前言" href="Text/Foreword.xhtml"/>
</guide>
```

以下代码显示了链接的*cover.xhtml*、*Copyright.xhtml*、*Foreword.xhtml*文件所代表的结构语义：

```
<nav epub:type="landmarks" id="landmarks" hidden="">
  <h1>Landmarks</h1>
    <ol>
      <li><a epub:type="toc" href="#toc">目录</a></li>
      <li><a epub:type="cover" href="../Text/Cover.xhtml">封面</a></li>
      <li><a epub:type="copyright-page" href="../Text/Copyright.xhtml">版权页</a></li>
      <li><a epub:type="foreword" href="../Text/Foreword.xhtml">前言</a></li>
  </ol>
</nav>
```

在ePub 3标准中，当你为XHTML文件添加结构语义之后，可以在*nav.xhtml*文件中，查看landmarks属性（如上）。关于`epub:type`属性，见附录一。

1.2.3 toc.ncx

OPF 2里定义的*toc.ncx*被ePub 3的导航文件*nav.xhtml*所取代了。为了向前兼容的目的，ePub电子出版物可以包含*toc.ncx*。

有时使用Sigil打开一本ePub 3标准的电子书，Sigil会提示你缺少ncx文件。（请注意，在Sigil-0.9.10版中仍然强制要求epub3拥有并支持NCX，以便尽可能向后生成兼容的epub。）遇到这种问题，可以使用Sigil的一个ePub 3工具。依次点击「工具→Epub3 Tools→Generate NCX from Nav」，Sigil会根据现有的Nav导航文件，生成一个ncx文件，修复缺少ncx文件的报错。另一种原因是，toc文件路径是绝对的，处理办法是删除路径信息，只保留*toc.ncx*值，见本节的ncx文件不存在。

由于ePub 2的ncx和ePub 3的导航文件采用不同的机制确定导航信息，它们可以在ePub 3电子出版物里不冲突地共存。

采用ePub 3标准会生成一个*nav.xhtml*导航文件，它不能被删除（除非该电子书降级为ePub 2标准，见8.5.3节）。当你选择了使用ePub 3标准，并且想对默认的*nav.xhtml*文件做页面设计时，请切记在目录确定后再对*nav.xhtml*文件做修改，不然在目录变动后，该网页会重归默认状态，即你在该网页中所做的设计都会丢失。

NCX的`<head>`标记中包含四个meta标签：

- uid：数字图书的惟一ID。该标签应该和OPF文件中的dc:identifier对应。
- depth：反映目录表中层次的深度。
- totalPageCount和maxPageNumber：仅用于纸质图书，保留0即可。

docTitle/text的内容是图书的标题，和OPF中的dc:title匹配。

navMap是ncx文件中最重要的部分，定义了图书的目录。navMap包含一个或多个navPoint标签。每个navPoint都要包含下列标签：

- playOrder属性，说明文档的阅读顺序。和OPF spine中itemref标签的顺序相同。
- navLabel/text标签，给出该章节的标题。通常是章的标题或者数字，如「第一章」，或者像这个例子一样——「1.4 几种常见的电子书格式」。
- content标签，它的src属性指向包含这些内容的物理资源。就是OPF manifest中声明的文件（也可使用片段标识符引用XHTML内容中的锚标签——比如`content.html#footnote1`）。
- 还可以有一个或多个navPoint标签。ncx使用嵌套的导航点表示层次结构的文档。

当前电子书的navMap信息（部分）：

```
<navMap>
  <navPoint id="navPoint1">
    <navLabel>
      <text>1 ePub概述</text>
    </navLabel>
    <content src="Text/Chapter1.xhtml" />
    <navPoint id="navPoint2">
      <navLabel>
        <text>1.1 什么是ePub？</text>
      </navLabel>
      <content src="Text/Chapter1-1.xhtml" />
      <navPoint id="navPoint3">
        <navLabel>
          <text>1.1.1 创建过程</text>
      </navLabel>
      <content src="Text/Chapter1-1.xhtml#create_epub" />
      </navPoint>
      <navPoint id="navPoint4">
        <navLabel>
          <text>1.1.2 使用技术</text>
```

```
      </navLabel>
      <content src="Text/Chapter1-1.xhtml#epub_technology" />
      </navPoint>
      <navPoint id="navPoint5">
        <navLabel>
          <text>1.1.3 规范文档</text>
      </navLabel>
      <content src="Text/Chapter1-1.xhtml#epub_Standard" />
      </navPoint>
    </navPoint>
    <navPoint id="navPoint6">
      <navLabel>
        <text>1.2 组成ePub的文件</text>
      </navLabel>
      <content src="Text/Chapter1-2.xhtml" />
      <navPoint id="navPoint7">
        <navLabel>
          <text>1.2.1 mimetype和container.xml</text>
      </navLabel>
      <content src="Text/Chapter1-2.xhtml#epub_mimetype_container" />
      </navPoint>
      <navPoint id="navPoint8">
        <navLabel>
          <text>1.2.2 content.opf</text>
      </navLabel>
      <content src="Text/Chapter1-2.xhtml#epub_content" />
      </navPoint>
      <navPoint id="navPoint9">
        <navLabel>
          <text>1.2.3 toc.ncx</text>
      </navLabel>
      <content src="Text/Chapter1-2.xhtml#epub_toc" />
      </navPoint>
      <navPoint id="navPoint10">
        <navLabel>
          <text>1.2.4 出版物资源文件</text>
      </navLabel>
      <content src="Text/Chapter1-2.xhtml#epub_publish" />
      </navPoint>
    </navPoint>
    <navPoint id="navPoint11">
      <navLabel>
        <text>1.3 制作完ePub的后续工作</text>
      </navLabel>
      <content src="Text/Chapter1-3.xhtml" />
    </navPoint>
```

```
    <navPoint id="navPoint12">
      <navLabel>
        <text>1.4 几种常见的电子书格式</text>
      </navLabel>
      <content src="Text/Chapter1-4.xhtml" />
    </navPoint>
  </navPoint>
</navMap>
```

实际上，上面的目录信息在「目录」视图中更加直观（在Sigil中，选择菜单项「视图→目录」）：

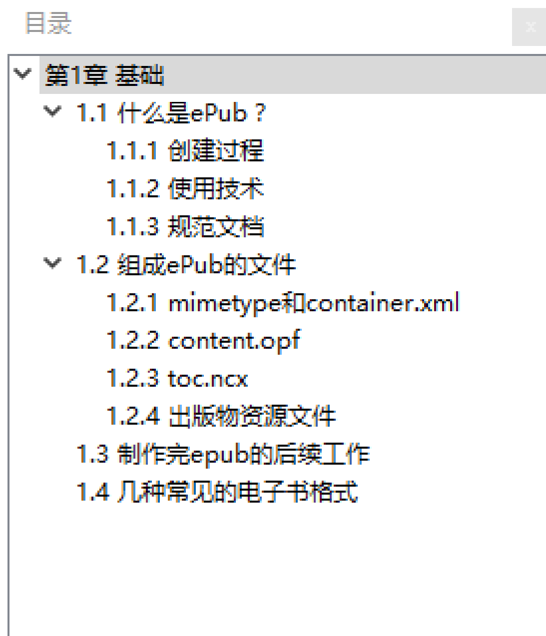


图1-2　Sigil的「目录」窗口

这个目录信息可以被阅读系统读取，辅助读者进行章节间的跳转。您还可以创建一个*TOC.xhtml*版本的目录，直接展示电子书结构而不像ncx一样只有用到时才可以看到。在ePub 3中，倒不需要这个文件，它的功能已经与*nav.xhtml*重合了。

作为XHTML内容文件，导航文件还利用CSS和隐藏属性为定制导航显示提供了灵活手段，同时还不影响阅读系统的信息访问。导航文档的结构和语义定义见ePub导航文件。

ncx和spine有什么不同？

两者很容易混淆，因为两个文件都描述了文档的顺序和内容。要说明两

者的区别，最简单的办法就是拿印刷书来打比方：OPF文件中的spine标签描述了书中的各个章节是如何实际连接起来的，比方说翻过第一章的最后一页就看到第二章的第一页（内容文档）。ncx在图书的一开始描述了目录。目录肯定会包含书中主要的章节，但是还可能包含没有单独分页的小节（内容文档中的小标题）。

一条法则是ncx包含的navPoint标签通常比OPF　spine中的itemref标签多。实际上，spine标签中的所有项都会出现在ncx中，但ncx可能更详细。

比较直观的做法是直接到OPF文件找到spine标签内容，并和ncx文件对比下。比如当前这本书的第一章，在spine标签中列出如下：

```
<spine toc="ncx">
  ...
  <itemref idref="Chapter1.xhtml"/>
  <itemref idref="Chapter1-1.xhtml"/>
  <itemref idref="Chapter1-2.xhtml"/>
  <itemref idref="Chapter1-3.xhtml"/>
  <itemref idref="Chapter1-4.xhtml"/>
  <itemref idref="Chapter1-5.xhtml"/>
  ...
</spine>
```

在spine标签中*Chapter1-1.xhtml*对应的是「1.1　什么是ePub？」，*Chapter1-2.xhtml*对应的是「1.2　组成ePub的文件」，以此类推。但是「1.1　什么是ePub？」这一节中还有3个小节，spine标签没有列出来，它列出来的只有Text文件夹中的内容文档。而「1.1.1　创建过程」、「1.1.2　使用技术」、「1.1.3　规范文档」3小节只存在于*Chapter1-1.xhtml*这个内容文档中，它们在*toc.ncx*文件中有列出来，见上文的示例代码。

上述内容来自使用ePub制作数字图书——基于XML的开放式eBook格式，感谢！

ncx文件不存在

打开一本ePub电子书时，如果提示如下：

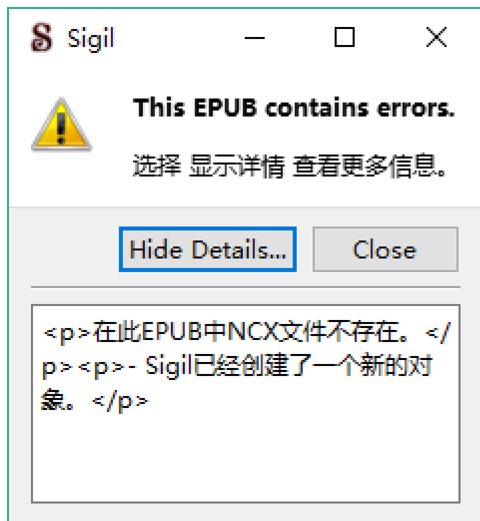


图1-3　打开epub文件报错：ncx不存在

点击按钮close，打开`content.opf`文件。在`<manifest>`标签内有两个引用资源代码：

```
<item href="toc.ncx" id="ncx" media-type="application/xhtml+xml" properties="nav" />
<item href="C%253A/Users/admin/AppData/Local/Temp/Sigil/scratchpad/f1f03513-39e4-47e1-b52d-eccb987d6d78/OEBPS/toc.ncx" id="ncx" media-type="application/x-dtbncx+xml" />
```

问题是出在第2个引用资源代码上，原因是引用了计算机上的资源且与上一条重复并且该文件是绝对路径存在的，处理办法是删除它。正确的文件路径信息：

```
<item href="toc.ncx" id="ncx" media-type="application/x-dtbncx+xml" />
```

然后查看`<spine>`标签是否引用了ncx目录，如果是下面这种：

```
<spine toc="">
```

则修改成如下：

```
<spine toc="ncx">
```

完成后保存ePub文件即可。

1.2.4　核心媒体类型

ePub的核心媒体类型在opf文件中可见一斑，下面分别介绍ePub核心媒体类型：

图像类型

所有的图片文件都需要放在Images文件夹里。ePub支持四种核心图像文件类型：

1. image/gif：常用于储存动态图片
2. image/jpeg：有损的图像格式
3. image/png：一般用于储存透明格式图片
4. image/svg+xml：可缩放矢量图形

虽然ePub定义了以上4种图像格式，但是还需要阅读系统的支持才能显示，比如*.gif*格式动态图片在多看阅读器中就无法正常播放。关于图像格式的知识见6.1节。

应用类型

ePub规范定义的XHTML文件类型是基于W3C[HTML5]，且继承了（如无另外规定）HTML5的全部的语义、结构和处理行为。可扩展超文本标记语言（eXtensible HyperText Markup Language，缩写为XHTML），是一种标记语言，表现方式与超文本标记语言（HTML）类似，不过语法上更加严格。

1. application/xhtml+xml：XHTML内容使用XHTML语法[HTML]的文档
2. application/javascript：脚本
3. application/x-dtbncx+xml：被取代的NCX
4. application/font-sfnt：OpenType和TrueType字体
5. application/font-woff：WOFF字体
6. application/smil+xml：媒体叠加文档
7. application/pls+xml：文本到语音（TTS）发音词典

音频类型

多看阅读、掌阅iReader、Gitden等均支持多媒体文件，ePub3支持嵌入在内容文档里面的音频、视频，并且继承这些标签提供的所有功能和特性。

1. audio/mpeg：MP3音频
2. audio/mp4：使用MP4容器的AAC LC音频

视频类型

ePub3目前没有将任何视频编解码器定义为核心媒体类型。有关ePub出版物中视频编解码器支持的信息性建议，请参阅EPUB出版物中的说明——阅读系统一致性。官方推荐使用H.264、VP8格式视频。

文字类型

层叠样式表，一种用来为内容文档添加样式（背景、边框、字体、间距和颜色等）的计算机语言，由W3C定义和维护。

1. text/css：CSS样式表

虽然ePub定义了CSS的相关规则，但是阅读系统支不支持是一回事，有的阅读系统只支持比较简单常见的CSS属性，有的则可能比较多。所以在制作一本电子书时，并不推荐实用不常用的CSS属性，这是因为即使你在Sigil中做的非常完美，但是却没有相应的阅读器去渲染实现，或者显示的乱七八糟，那根本没有意义。这是比较尴尬的事。关于阅读器的选择，见1.3节。

字体类型

阅读系统内置的字体有时无法满足我们的要求，比如对汉字生僻字（一般指扩展B平面以上的）的显示，此时需要用`@font-face`语法重新定义一种字体。`@font-face`语法的使用需要获得阅读系统的支持。

1. font/woff2：WOFF2字体

嵌入字体实现的手段，见8.3、8.4、8.5三节。

1.3 选择一个阅读器

虽然ePub标准并不难，但其XML文件必须符合特定的规范，常用的两种验证手段见8.2和8.5.4二节。

经过验证的ePub电子书，还需要在真机的阅读器中测试效果。鼓励你用更多的阅读器测试验证你的ePub，这不光可以检测电子书在阅读设备上的浏览效果，也可以帮助你提高制作水平。

移动端

如果你是移动端阅读，推荐使用以下APP来打开ePub文件：

表2　移动端ePub阅读器推荐

名称	logo	支持格式	规范支持	平台
多看阅读		TXT/ePub/PDF	★★	Andriod/iOS/Kindle
掌阅iReader		TXT/ePub/PDF	★☆	Andriod/iOS
Gitden Reader		ePub	★★★	Android/iOS
Lithium		ePub	★★☆	Android

以上4款阅读器是目前应用市场比较好的阅读器，其中掌阅和多看是国产的阅读器，虽然并不能完全支持ePub 3，但是流畅度、用户界面、书城等功能很适合国人；Gitden Reader和Lithium两款阅读器是国外团队开发的，支持ePub 3规范的比较多，适合作为本地阅读器使用。作者本人常用的阅读器是多看阅读，也推荐大家使用这款阅读器。

PC端

在PC端阅读ePub文件，有Windows 10的Edge浏览器、NeatReader、Chrome浏览器的扩展程序Readium，软件的话就是Calibre、Sigil、AZARDI这些了。

如果你需要编辑ePub文件，推荐使用Sigil。关于Sigil的使用指南，见第

5章。

你也可以对ePub文件重命名，改文件类型epub为zip。使用解压软件解压，就可以看到ePub的所有组成文件，根据文件类型选择编辑程序进行编辑。除了图像、音频、视频、字体等，其他如xhtml、html、css、svg、opf、ncx、xml之类的文件，均可以用Windows系统的记事本程序编辑。

1.4　几种常见的电子书格式

电子书是书籍的数字媒体表现形式，它的特点是便于携带、使用方便、而且占用空间很小，在移动设备（例如专用阅读设备、手机、平板电脑）上可以大量存储，不用占用太多实体空间，非常适合快节奏的现代生活。另外，数字技术的发展使得电子出版物的版权问题得到了很好的解决。在网络如此便捷的今天，电子书用户能够用更少的钱去购买更多的书籍，价格优势也是很多用户选择电子书的一个重要原因。电子书格式也是丰富多彩，各有所长。常见格式包括TXT、PDF、EPUB、CAJ等等。

在不同的设备上最适合阅读的电子书格式不同，请根据现有的阅读设备选择一种格式。下面内容介绍常见的几种电子书格式：

PDF

PDF，这是最常见的电子书格式之一，标准由Adobe主导。PDF格式的电子书在网上找起来非常容易，很多学术资料只有PDF格式的，而且几乎所有的平台都支持PDF，图表、插图、多字体等各种各样的复杂排版对于PDF来说都不在话下，它还可以保持纸书原有的样貌，这让PDF成为了一个比较主流的电子书格式。PDF有扫描版与文字版之分，文字版无须在意清晰度的问题，但是一些比较古老的扫描版PDF，往往清晰度很差，扫描版PDF每一页实际上是一张类似图片的文件。

PDF文件有三种类型：图像PDF、双层PDF、版式还原的纯文本单层PDF。双层PDF，是指用户浏览看到的是图像层，在全文检索和文字拷贝时使用的是文字层，即为双层；版式还原的纯文本单层PDF，是指全部文字是以纯文本形式展现，图片以图像形式展现，再按照图书的版式重新排版的PDF。借助Sigil插件我们可以把ePub电子书转换成PDF文件，见8.6节。

创建PDF的工具有Adobe InDesign、方正书版、Microsoft Office Word等。

PC端：Adobe Acrobat、Adobe Reader、WPS、Chrome浏览器

移动端：多看阅读（PDF重排）

ePub

ePub，当下最流行的电子书格式之一，它是一种开源格式，这意味着它自身的标准都是有争议的。ePub除了支持插图、字体等复杂排版功能外，甚至还可以插入音频与视频，而且它可以自适应屏幕，PDF一般是无法自适应屏幕的，所以用一个6寸的设备去阅读8寸的PDF总是会有点捉襟见肘（需要不时的放大缩小），但ePub就不会出现这些情况，无论你使用4寸或者10寸的设备，都能得到一个不错的阅读体验。

PC端：Edge浏览器、Sigil、Icecream Ebook Reader、Calibre

移动端：多看阅读、掌阅iReader、Gitden Reader、Lithium、Google Book、iBooks

Mobi和Azw3

Mobi和Azw3，这两个电子书格式都是由亚马逊这一巨头支撑的，这两种格式与ePub单从阅读体验上来说相差无几，但Mobi对复杂排版的支持度相比ePub弱了很多，新推出的Azw3也支持更多的复杂排版，与ePub可以相提并论，但是Azw3很封闭，基本上全是由亚马逊一家支撑，而ePub是开放的。由于亚马逊庞大的电子书市场以及Kindle在电子书阅读器里的垄断地位，Mobi与Azw3的市场占有率也很高。借助Sigil插件我们可以把Azw3转换成ePub格式，见12.5节。

PC端：Kindle Previewer、Calibre

移动端：Kindle App

你也可以使用Kindle电纸书设备阅读此类格式。

TXT

TXT，这可以说是最基础的格式，大部分人从一开始接触的电子书应该就是TXT，TXT很流行，很多书都可以在网上找到TXT版的，但是现在来看，TXT作为电子书是很简陋的，不支持复杂排版、不可以插入图片、没有导航目录。一本排版精美的电子书所需要的要素基本全不具备，所以现在用TXT来看的人已经越来越少了。在PC端很多软件都可以打开TXT文本，如Windows系统自带的记事本程序，专门处理文本的EmEditor等；在移动端，常见的阅读器，如多看阅读、掌阅等均可以打开TXT文件，并且支持特定内容的章节目录提取、自定义样式等效果。

其他

CHM与EXE，这两种都是给电脑量身定制的，随着手机与电子书阅读器的发展，也逐渐的淡出视线。在Windows系统中可以直接打开，不需要其他软件。

Djvu，是一种高清扫描文件格式，比较罕见，常见于扫描古籍，需要用特定阅读器打开，如WinDjView。

Uvz，一种类似图像PDF的文件，习惯上我会把文件类型uvz改成zip解压，然后使用Pdg2Pic制作成PDF格式。可以使用老马的UnicornViewer软件打开。

Caj，知网上下载的论文大多是这个格式的，需要用特定阅读器打开，如CAJViewer。

PDG，超星设计的电子书标准，用超星阅读器打开，也可以用软件转换成PDF（图像格式）。转换软件推荐使用老马的Pdg2Pic软件。此种格式，可以去老马的博客学习了解下。

Docx，是Microsoft Office2007之后版本使用的，用新的基于XML的压缩文件格式取代了其目前专有的默认文件格式，在传统的文件名扩展名后面添加了字母「x」（即「.docx」取代「.doc」、「.xlsx」取代「.xls」、「.pptx」取代「.ppt」）。

1.5 无障碍访问

ePub的一个主要目标是使内容访问无障碍，ePub 3有好些功能支持这一需求。本节对一些成熟的最佳做法做详细介绍，确保在不同的场景中都能完整显示ePub出版物。

值得注意的是，虽然访问无障碍本身是重要的，能够被访问的内容也是更有价值的内容：一个无访问障碍的出版物，不管是全部或部分，也不管是通过人工还是自动化工作流程，要适应更多的设备和更易于重复使用。

1.5.1 导航

ePub 3增加导航文件对ncx文件进行改进。如前面内容文件所指出的，由这些提供的新功能，代表的是一个更通用、更灵活的导航系统。

对文档进行导航的需要不只是一个访问无障碍的问题。相反，一个完整的导航文件是让所有用户都能够轻松、充分地访问文件内容。作为结果，首要的nav元素（定义请看The toc nav Element ContentDocs）应该充分反映出版物的结构。

对于高度结构化的文件，可能不希望显示完整的内容给用户。可以使用HTML5的hidden属性修改显示级别，避免减少辅助阅读系统所能够获得的信息。

如果出版物包含非结构化的兴趣点，如数字、表格等，也鼓励作者提供额外的nav元素，以进一步提高对内容的访问。

1.5.2 语义标记

HTML 5支持一些旨在使标记的语义更清晰（如section，nav，aside）的新元素，又为几个HTML 4的元素增加了定义更明确的语义。这些元素，加上创作具有良好结构的Web内容的最佳做法，应在创建ePub的

XHTML内容文件时加以使用。这些新增功能允许将内容更好地分组和定义，既为展示文件的结构，也方便它们的逻辑导航。XHTML内容文件也本地支持对ARIA角色、状态属性和事件的引入，提高了辅助技术与内容交互的能力。

ePub 3进一步引入epub:type属性，以在功能上等同于W3C的角色属性。此属性允许XHTML内容文档中的任何元素都可以使用受控词汇、术语来添加关于其目的、意义的额外信息。更多信息请参考Semantic Inflection ContentDocs。

1.5.3 动态布局

ePub的设计中心是动态布局：总希望在空白处对内容布局，而不是以事先分页的方式排版（即预期某个具体尺寸的「页」）。这一核心能力是非常有用的，例如，用于在不同大小的设备屏幕或窗口尺寸上做最佳渲染，它也方便和简化了内容的访问。

虽然加入格式化程度更高的ePub内容是可行的——例如通过bmp图像或SVG图形，甚至是使用CSS明确定位或表格元素以实现特定的视觉布局——但还是要极力劝阻作者使用这种技术。它们在ePub是不可靠的，原因是许多阅读系统是按照分页的方式来渲染内容，而不是建立一个可以上下滚动的视窗，也是因为每个阅读系统的分页算法可能是自定义的。如果传递出版物的内容（例如漫画）需要使用这些技术，则应该总是包括备选方案。

一般而言，最好是使用ePub样式表来实现视觉的丰富样式，避免使用绝对尺寸或定位。

1.5.4 有声播报和媒体叠加

内容的有声播报对访问无障碍是重要的，对许多其他用户也是个满意的功能。提供有声播报的底线是利用为动态布局设计的语义化HTML。关于如何使用ePub XHTML文档内本地设施的详细信息，请参阅语音合成。

媒体叠加提供了将出版物的文本和音频内容同步的功能。媒体叠加的功用超出了辅助领域：比如，作为练习阅读能力的工具，文字和音频的同步在许多情况下是有用的。

1.5.5 备选方案

并非所有的格式在其本地格式下都是可访问的，也并不是所有的用户都喜欢按照所提供的默认格式来阅读。ePub定义了提供备选方案的各种手段，使出版物在这些情况下的选择性渲染成为可能。

出版物和内容级的备选方案定义在限制和备选。这些允许对出版物的具体资源，如SVG图像或视频剪辑，进行可选渲染。

此外，一个完整工作的多个实例可以发表在单个出版物，其方法是在OCF容器文件里面定义多个rootfile元素（详见Container – META-INF/container.xml OCF3）。这样的备选方案可以用来，例如，使得由一系列SVG页面组成的连环画小说，能够附带一个按照XHTML定义、可访问的文本版。

1.5.6 脚本

ePub3对脚本内容进行逐渐改进的处理，并且不得影响脚本文件的完整性（即在脚本不可用时，不得导致信息丢失）。因此，虽然使用了脚本的文件可能会提供备选方案以利于对其内容的访问，但是，没有这些，文件也必须能够被访问。

好几个ePub机制可以进一步最大程度降低和限制出版物里面的脚本，以改善可访问性：

1. ePub HTML5新增加的声明性触发元素，使图像或文本元素能够作为音频和视频的控制（例如，开始、停止和暂停播放）。此元素让为实现类似功能而屡屡使用的脚本可以寿终正寝了。
2. mediaType元素对以脚本为基础的支持提供了封装手段，用于渲染常用的XML词汇或其它常用的内容类型、也是用于生成不过时的出版物，以备将来的阅读系统在本地支持这些类型的内容。
3. 由type属性提供的语义变化功能使作者能够给阅读系统提供关于内容属性的提示。一个实用例子是定义图像、视频等元素拥有可缩放的属性值，在这种情况下阅读系统可为用户提供访问放大画面的手段，而在正常布局情况下则会超界。这种翻转效果通常是借助Web上下文的脚本来实现的，但阅读系统产生的布局何其多，实施脚本谈何容易。
4. switch element元素提供了声明性质的手段，作者用它为用户剪裁显示内容，而无需诉诸脚本化的解决方案。

应经常借鉴Web文档的那些用于脚本访问的最佳做法，如WAI-ARIA里面提供的。应把脚本的使用保留给那些交互式用户体验必不可少的情形。

第2章

HTML知识

超文本标记语言（HyperText Markup Language，缩写为HTML）是一种用于创建网页的标准标记语言。HTML是一种基础技术，常与CSS（网页的表现与展示效果）、JavaScript（功能与行为）一起被众多网站用于设计网页、网页应用程序以及移动应用程序的用户界面。

HTML元素是构建web网站的基石，它定义了网页内容的含义和结构。HTML可以嵌入图像，结构化信息——例如标题、段落和列表等等，也可用来在一定程度上描述文档的外观和语义（浏览器样式）。HTML的语言形式为尖括号包围的HTML元素（如`<html>`），浏览器使用HTML标签和脚本来诠释网页内容，但不会将它们显示在页面上。

维护HTML和CSS标准的组织万维网联盟（W3C）鼓励人们使用CSS替代一些用于表现的HTML元素，如加粗、斜体、强调等。

HTML中的标签是不区分大小写的。也就是说，当你输入标签时，你既可以使用大写字母也可以使用小写字母。例如，标签`<title>`可以写作`<title>`，`<TITLE>`，`<Title>`，`<TiTlE>`，等等，它们都可以正常工作。不过，从一致性、可读性和其他各方面来说，最好仅使用小写字母。

一个HTML元素构成如下（这里以段落元素为例）：



图2-1　HTML元素构成

主要部分是：

1. 开始标签（Opening tag）：包括元素的名称（在本例中，p），包裹在开始和结束尖括号中。这表示元素开始或开始生效——在这种情况下，表示了一个段落的开头。
2. 结束标签（Closing tag）：这与开始标记相同，除了它在元素名称之前包含正斜杠。这表示元素结束的位置——在这种情况下，表示了一个段落的结尾。没有包含结束标记是一个常见的初学者错误，并可能导致奇怪的结果。
3. 内容（Content）：这是元素的内容，在这种情况下只是文本。
4. 元素（Element）：开始标记、加结束标记、加内容，等于元素。

不是所有元素都拥有开始标签、内容和结束标记。一些元素只有一个标签，通常用来在此元素所在位置插入或嵌入一些东西。例如：元素`<img>`是用来在元素`<img>`所在位置插入一张指定的图片。这些元素称作自闭合（self-closing）元素，如：

- `<meta>`标签：设置页面元信息
- `<br>`标签：换行
- `<hr>`标签：水平线
- `<img>`标签：图像

- <link>标签：定义文档与外部资源的关系的链接
- <source>标签：为媒体元素定义媒体资源
- ……

可以为HTML元素添加属性，通常用英文双引号包裹。

如同大部分的编程语言一样，在HTML中有一种可用的机制来在代码中书写注释——注释是被浏览器忽略，而且是对用户不可见的，它们的目的是允许你描述你的代码是如何工作的和不同部分的代码做了什么等等。如果你在半年后重新查看你的代码库，而且不能记起你所做的事情——或者当你处理别人的代码的时候，那么注释是很有用的。

为了将一段HTML中的内容置为注释，你需要将其用特殊的记号<!--和-->包括起来，比如：

```
<p>我没有在注释里。</p>
<!-- <p>我在！</p> -->
```

2.1 块状元素

块状元素（Block element），字面理解就是一个方块，它一般是其它元素的容器元素，可以容纳块状元素和行内元素，默认是不会和其它元素在同一行，在显示上是垂直布局。最常用块状元素是<div>和<p>。特点：

- 总是在新行上开始；
- 高度，行高以及外边距和内边距都可控制；
- 宽度缺省是它的容器的100%，除非设定一个宽度；
- 它可以容纳内联元素和其他块元素。

大部分的文本结构由标题和段落组成，不管是小说、报刊、教科书还是杂志等。内容结构化会使读者的阅读体验更轻松，更愉快。另外在我们身边的任何地方都要依赖语义学——我们依靠以前的经验就知道日常事物都代表什么；当我们看到什么，我们就会知道它代表什么。举个例子，我们知道红色交通灯表示「停止」，绿色交通灯表示「通行」。如果运用了错误的语义，事情会迅速地变得非常棘手 (难道有某个国家使用红色代表通行？我不希望如此)。那么，在一本书中不同类型的内容就要适当地添加各自的HTML标签。

2.1.1 p（段落）

ePub中，常使用<p>标签包裹正文段落。你可以使用单标签选择器给p元素设置样式，也可以通过给所有正文添加class来设置正文样式。

p元素会自动在其前后创建一些空白。浏览器会自动添加这些空间（浏览器默认样式），您也可以在样式表中重新定义。对<p>标签定义上下间距，即设置排版中的段落间距。段落间距设置成什么值合适，会在后面的章节中介绍。

下面是3个案例展示了p元素的某些规则。

简单的段落

此例演示段落元素中的文字如何被浏览器显示。

```
<p>这是一个简单的正文段落。</p>
<p>段落元素由p元素定义。</p>
```

显示如下：

这是一个简单的正文段落。

段落元素由p元素定义。

更多的段落

本例演示段落元素的某些默认的处理行为。

```
<p>
这个段落
在源代码中
包含许多行
但是浏览器
忽略了它们。
</p>
<p>
这个段落
在源代码          中
包含     许多行
但是          浏览器
忽略了   它们。
</p>
<p>
段落的行数依赖于浏览器窗口的大小。如果调节浏览器窗口的大小，将改变段落中的行数。
</p>
```

显示如下：

这个段落 在源代码中 包含许多行 但是浏览器 忽略了它们。

这个段落 在源代码 中 包含 许多行 但是 浏览器 忽略了 它们。

段落的行数依赖于浏览器窗口的大小。如果调节浏览器窗口的大小，将改变段落中的行数。

在这个案例中，浏览器忽略了字与字之间的空格，在最终的显示中，只显示了一个空格。多行的段落之间，同样是用空格代替。

「诗歌」问题

本例演示HTML格式化的某些问题。

```
<p>
	春眠不觉晓，
		处处闻啼鸟。
			夜来风雨声，
				花落知多少。
</p>
<p>注意，浏览器忽略了源代码中的排版（省略了多余的空格和换行）。</p>
```

显示结果如下：

春眠不觉晓， 处处闻啼鸟。 夜来风雨声， 花落知多少。

注意，浏览器忽略了源代码中的排版（省略了多余的空格和换行）。

若想做到代码案例中的效果，需要使用pre元素。

2.1.2 div（分区或节）

<div>可定义文档中的分区或节（division/section）。

<div>标签可以把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。

如果用id或class来标记<div>，那么该标签的作用会变得更加有效。

<div>是一个块级元素。这意味着它的内容自动地从一个新行开始。实际上，换行是<div>固有的唯一格式表现。可以通过<div>的class或id应用额外的样式。

不必为每一个<div>都加上class或id，虽然这样做也有一定的好处。

可以对同一个<div>元素应用class或id属性，但是更常见的情况是只应用其中一种。这两者的主要差异是，class用于元素组（类似的元素，或者可以理解为某一类元素），而id用于标识单独的唯一的元素。

> 请使用<div>标签来组合块级元素，这样就可以使用样式对它们进行格式化。

在图书排版中，我们可以使用<div>标签组织插图或几段密切相关的段落，如诗歌、书信、图片格式的表格等。

演示案例：

```
<div class="pictures">
  <p>图片标题</p>
  <img alt="img001" src="../Images/img001.jpg"/>
  <p>图片说明</p>
</div>
```

上面这段HTML模拟了一个插图的结构。其中div元素把每个插图的标题、图片和说明组合在一起，也就是说，div元素为文档添加了额外的结构。同时，由于这些div属于同一类元素，所以可以使用`class="pictures"`对这些div元素进行标识，这么做不仅为div元素添加了合适的语义，而且便于进一步使用样式对div元素进行格式化，可谓一举两得。我们还可以为div元素里的图片标题、图片说明添加class，然后进行格式化，使之排版效果更佳。

2.1.3　h1~h6（标题）

<h1> ~ <h6>标签可定义标题，<h1>定义最大的标题，<h6>定义最小的标题。

由于<h1> ~ <h6>元素拥有确切的语义，在制作电子书的过程中慎重地选择恰当的标题层级来构建文档的结构，可以依据原书选择对应的标题层级（电子书目录的生成即是依靠标题层级的）。因此，请不要利用标题标签来改变同一行中的字体大小。相反，我们应当使用CSS样式来达到格式化的效果。

六个不同的HTML标题（从大到小）：

```
<h1>这是1级标题</h1>
<h2>这是2级标题</h2>
<h3>这是3级标题</h3>
```

```
<h4>这是4级标题</h4>
<h5>这是5级标题</h5>
<h6>这是6级标题</h6>
```

因为这本书已经针对标题进行了格式化，显示效果与浏览器默认样式大不一样。或者到W3School查看。应用了标题标签的文本显示如下：

这是1级标题

这是2级标题

这是3级标题

这是4级标题

这是5级标题

这是6级标题

图2-2　浏览器默认显示的各层级标题

在Sigil中快速更改光标所在段落的样式可以使用工具栏中的标题按钮组。见5.4.8节内容。

2.1.4　table（表格）

`<table>`标签定义HTML表格。自制电子书时建议使用`<table>`标签来显示表格内容（小型表格内容），而不是直接用图片代替。对于大型表格，建议拆分为小表格，再使用分页属性分屏显示。在支持ePub 3规范的阅读器或专业的电纸书设备Kindle中，双击表格内容，可以对表格进行放大、移动等操作。

简单的HTML表格由table元素以及一个或多个tr、th或td元素组成。tr元素定义表格行，th元素定义表头，td元素定义表格单元。

更复杂的HTML表格也可能包括caption、col、colgroup、thead、tfoot以及tbody元素。

用`<thead>`标签包裹表格标题，`<tfoot>`包裹表格注释，`<tbody>`包裹表格内

容。虽然<tfoot>标签在<table>表格中的位置不重要，无论在什么位置，在Sigil、Chrome、Lithium、Gitden等软件中，一律会默认把<tfoot>标签显示在表格底部。但是在多看阅读器中，除非<tfoot>标签在表格底部，不然它会根据所在位置显示<tfoot>标签里的内容。所以在需要用到<tfoot>标签时，记得把它移动到表格底部。

一个简单的HTML表格，包含两行两列：

```
<table border="1">
 <tbody>
  <tr>
    <th>月份</th>
    <th>天数</th>
  </tr>
  <tr>
    <td>一月</td>
    <td>31天</td>
  </tr>
 </tbody>
</table>
```

显示如下：

月份	天数
一月	31天

上述中<th>标签表示表格的表头（与td元素类似，区别在字体默认样式是加粗且水平居中的），<td>标签表示一个单元格的内容。table可选的属性可以参考W3School网站，从实用角度出发，最好不要规定属性，而是使用CSS来添加样式。

横跨行和横跨列：

```
<table border="1">
  <thead>
    <tr>
      <th colspan="4">演示如何合并单元格</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="3">一</td>
      <td>二</td>
      <td>三</td>
      <td>四</td>
```

```
    </tr>
    <tr>
      <td>五</td>
      <td>六</td>
      <td>七</td>
    </tr>
    <tr>
      <td>八</td>
      <td>九</td>
      <td>十</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="4">备注：这个是表格的脚注或表注</td>
    </tr>
  </tfoot>
</table>
```

显示结果如下：

演示如何合并单元格

一	二	三	四
五	六	七	
八	九	十	
备注：这个是表格的脚注或表注			

<tfoot>标签定义表格的页脚（脚注或表注）。该标签用于组合HTML表格中的表注内容。

tfoot元素应该与thead和tbody元素结合起来使用。thead元素用于对HTML表格中的表头内容进行分组，而tbody元素用于对HTML表格中的主体内容进行分组。

> 如果您使用thead、tfoot以及tbody元素，您就必须使用全部的元素。它们的出现次序是：thead、tbody、tfoot，这样浏览器就可以在收到所有数据前呈现页脚了。您必须在table元素内部使用这些标签。

thead、tfoot以及tbody元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

其中，border=""如果不设置的话，表格边框为无，即只能看到内容，而看不到边框线条。然后如何把单元格边框进行合并，需要用到CSS样式，后面章节中再讲。如果想要把表格宽度适应预览页面宽度，只需设置width:100%;即可。

从显示效果上可以看到表头「演示如何合并单元格」横向独占4列，即4个单元格；「一」纵向独占3行，即3个单元格。

HTML表格有两类单元格：表头单元-包含头部信息（由th元素创建）；标准单元-包含数据（由td元素创建）。

<td>元素中的文本一般显示为正常字体且左对齐。

> 有些表格比较复杂，需要合并行或者合并列，这就需要用到colspan=""（规定单元格可横跨的列数。col＝column　竖列）和rowspan=""（规定单元格可横跨的行数。row＝row　横行）两个属性。横跨列或行后，需删除多余<td>单元格。
> 多看阅读中，如果表格某个单元格与当前页面单元格加起来的高度超过了设备的屏幕高度，则该单元格及其以下的所有单元格都不会显示出来（应该是个bug）。

2.1.5　ol、ul（列表）

标签定义有序列表（ordered lists）。所谓有序列表，指的是会对列表对象按照某种顺序进行自动编号，如1、2、3……，a、b、c……等等。

有序HTML列表

```
<ol>
  <li>咖啡</li>
  <li>牛奶</li>
  <li>茶</li>
</ol>
<ol start="50">
  <li>咖啡</li>
  <li>牛奶</li>
  <li>茶</li>
</ol>
```

这两个列表的显示效果如下：

1. 咖啡
2. 牛奶
3. 茶

1. 咖啡

2. 牛奶
3. 茶

其中start规定有序列表的起始值，如果你的列表标记类型是「lower-alpha（小写英文字母）」或非数字的话，你想要第一个列表的起始编号是c，则start值应该是3而不是c。

不同类型的有序列表

有序列表如果不设置type属性，默认的编号是以数字为标记类型。在有序列表中使用type属性：

```
<p>数字列表：</p>
<ol>
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ol>
<p>字母列表：</p>
<ol type="A">
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ol>
<p>小写字母列表：</p>
<ol type="a">
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ol>
<p>罗马字母列表：</p>
<ol type="I">
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ol>
<p>小写罗马字母列表：</p>
<ol type="i">
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
```

```
</ol>
```

显示结果如下：

数字列表：

1. 苹果
2. 香蕉
3. 柠檬
4. 桔子

字母列表：

A. 苹果
B. 香蕉
C. 柠檬
D. 桔子

小写字母列表：

a. 苹果
b. 香蕉
c. 柠檬
d. 桔子

罗马字母列表：

I. 苹果
II. 香蕉
III. 柠檬
IV. 桔子

小写罗马字母列表：

i. 苹果
ii. 香蕉
iii. 柠檬
iv. 桔子

尽管不赞成使用type属性，不过所有浏览器都支持type属性。建议使用CSS的list-style-type属性代替type。这个在后面章节中会说到。

无序HTML列表

标签定义无序列表（unordered lists）。无序列表指的是列表不会按照某种顺序，从第一个到最后一个进行某种标记类型的排序。一个无序列表：

```
<ul>
```

```
  <li>咖啡</li>
  <li>茶</li>
  <li>牛奶</li>
</ul>
```

显示效果如下：

- 咖啡
- 茶
- 牛奶

可设置的type属性有disc（实心圆）、square（实心方块）、circle（空心圆）。建议使用CSS的list-style-type属性代替type。不同类型的无序列表：

```
<p>Disc项目符号列表：</p>
<ul type="disc">
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ul>
<p>Circle项目符号列表：</p>
<ul type="circle">
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ul>
<p>Square项目符号列表：</p>
<ul type="square">
  <li>苹果</li>
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ul>
```

显示结果如下：

Disc项目符号列表：

- 苹果
- 香蕉
- 柠檬
- 桔子

Circle项目符号列表：

- 苹果
- 香蕉
- 柠檬
- 桔子

Square项目符号列表：

- 苹果
- 香蕉
- 柠檬
- 桔子

如何嵌套列表

和<ol>可以互相嵌套。测试代码：

```
<ul>
  <li>咖啡</li>
  <li>茶
    <ul>
      <li>红茶</li>
      <li>绿茶</li>
    </ul>
  </li>
  <li>牛奶</li>
</ul>
```

显示效果如下：

- 咖啡
- 茶
    - 红茶
    - 绿茶
- 牛奶

其中嵌套的列表，type属性自动与主列表做了区分。不过在多看阅读和掌阅iReader中，嵌套列表type属性与主列表type属性一样，未做区分，都是实心圆。有序列表中也可以嵌套无序列表：

1. 咖啡
2. 茶
    - 红茶

    - 绿茶
5. 牛奶

更复杂的嵌套列表4中阅读器都支持，3层列表：

- 咖啡
- 茶
    - 红茶
    - 绿茶
        - 中国茶
        - 非洲茶
- 牛奶

与双层嵌套列表一样，嵌套的列表之间的标记类型均自动做了区分，但在多看阅读和掌阅iReader中，嵌套列表type属性与主列表type属性一样，未做区分，都是实心圆。

2.1.6 blockquote（引用）

<blockquote>标签定义块引用。<blockquote>与</blockquote>之间的所有文本都会从常规文本中分离出来，经常会在左、右两边进行缩进（增加外边距），而且有时会使用斜体。也就是说，块引用拥有它们自己的空间。

为什么使用blockquote元素呢？我们在排书时，会经常遇到正文段落中引用了别的书籍的一部分内容。这部分内容我们一般会与正文做区分，用<div>标签包裹一段或多段引用，然后给div元素分配class或id，如<div class="quote">，这样做的话比直接使用<blockquote>标签要多输入字符。如果我们不喜欢<blockquote>标签默认的浏览器样式，直接用CSS选择器选择blockquote元素，进行样式的编写即可。既语义明确，又便于维护。

标记长的引用：

```
<blockquote>
  When authoring ePub 3 book contents, in order to place footnote number within the text and to show the footnote contents in the pop-up window, follow the processes as following.
</blockquote>
```

显示效果：

> When authoring ePub 3 book contents, in order to place footnote number within the text and to show the footnote contents in the pop-up window, follow the processes as following.

多段引用，`<blockquote>`标签也可以包含块级元素p。

2.1.7 section（节）

`<section>`标签定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。section元素与blockquote元素类似，它是HTML 5中的新标签。

section元素用来对网站或应用程序中页面上的内容进行分块，一个section元素通常由内容及其标题组成。但section元素并非一个普通的容器元素；当一个容器需要直接定义样式或通过脚本定义行为时，推荐使用div元素而非section。比如：

```
<section>
  <h1>苹果</h1>
  <p><b>苹果</b>，植物类水果，多次花果……（「苹果」文章正文）</p>
</section>
```

section元素的作用是对页面上的内容进行分块，或者说对文章进行分段，不要将它与表示「有着自己的完整的、独立的内容」的article元素混淆。

下面来看两个article元素与section元素结合使用的示例，希望能够帮助你更好地理解article元素与section元素的区别。

```
<article>
  <h1>苹果</h1>
  <p><b>苹果</b>，植物类水果，多次花果……</p>
  <section>
    <h2>红富士</h2>
    <p>红富士是从普通富士的芽（枝）变中选育出的着色系富的统称……</p>
  </section>
  <section>
    <h2>国光</h2>
    <p>国光苹果品，又名小国光、万寿。原产美国，1600年发现的偶然实生苗……</p>
  </section>
</article>
```

代码中的内容首先是一段独立的、完整的内容，因此使用article元素。

该内容是一篇关于苹果的文章，该文章分为3段，每一段都有一个独立的标题，因此使用了两个section元素。记住，对文章分段的工作也是使用section元素完成的。

接下来看一个包含article元素的section元素的示例：

```
<section>
  <h1>水果</h1>
  <article>
    <h2>苹果</h2>
    <p>苹果，植物类水果，多次花果……</p>
  </article>
  <article>
    <h2>橘子</h2>
    <p>橘子，是芸香科柑橘属的一种水果……</p>
  </article>
  <article>
    <h2>香蕉</h2>
    <p>香蕉，属于芭蕉科芭蕉属植物，又指其果实，热带地区广泛栽培食用……
</p>
  </article>
</section>
```

这个示例比前面的示例复杂了一些，首先，它是一篇文章中的一段，因此没有使用article元素。但是，在这一段中，可以分为几块独立的内容，因此嵌入了几个独立的article元素。

看到这里，你可能又糊涂了，这两个元素可以互换使用吗？它们的区别到底是什么呢？事实上，在HTML 5中，article元素可以看成是一种特殊种类的section元素，它比section元素更强调独立性。section元素强调分段或分块，而article强调独立性，具体来说，如果一块内容相对来说比较独立、完整的时候，应该使用article元素，但是如果想将一块内容分成几段的时候，应该使用section元素进行分段。另外，在HTML 5中，div元素变成了一种容器，当使用CSS样式的时候，可以对这个容器进行一个总体的CSS样式的套用。

最后，关于section元素的使用禁忌总结如下：

- 不要将section元素用作设置样式的页面容器，因为那是div元素的工作。
- 如果article元素、aside元素或nav元素更符合状况，不要使用section元素。
- 不要为没有标题的内容区块使用section元素。

> section不是通用容器元素。如果仅仅是用于设置样式或脚本处理，应用div元素。一条简单的准则是，只有元素内容会被列在文档大纲中时，才适合用section元素。

2.1.8 figure（流内容）

在介绍<div>标签时，我举了一个插图的例子，为了定义这个插图的样式，我对div元素添加了class="pictures"。为什么起这个名字？它可以在后期维护时看一眼就能明白这个<div>标签的含义，即增强了语义化。现在我们可以直接用figure元素，进行插图的组织。

<figure>标签规定独立的流内容（图像、图表、照片、代码等等）。figure元素的内容应该与主内容相关，但如果被删除，则不应对文档流产生影响。

```
<p>上海卢浦大桥是当今世界第一钢结构拱桥，是世界上跨度最大的拱形桥。它也是世界上首座完全采用焊接工艺连接的大型拱桥（除合拢接口采用栓接外），现场焊接焊缝总长度达4万多米，接近上海市内环高架路的总长度。卢浦大桥像澳大利亚悉尼的海湾大桥一样具有旅游观光的功能。</p>
<figure>
  <p>黄浦江上的卢浦大桥</p>
  <p>拍摄者：W3School 项目组，拍摄时间：2010年10月</p>
  <img alt="shanghai_lupu_bridge" src="../Images/shanghai_lupu_bridge.jpg"/>
</figure>
```

测试效果：

上海卢浦大桥是当今世界第一钢结构拱桥，是世界上跨度最大的拱形桥。它也是世界上首座完全采用焊接工艺连接的大型拱桥（除合拢接口采用栓接外），现场焊接焊缝总长度达4万多米，接近上海市内环高架路的总长度。卢浦大桥像澳大利亚悉尼的海湾大桥一样具有旅游观光的功能。



黄浦江上的卢浦大桥

拍摄者：W3School 项目组，拍摄时间：2010年10月

<figure>与</figure>标签中的内容会从正文中分离出来，四周的边距也

会自动添加。

> 请使用<figcaption>元素为figure添加标题（caption）。如：

```
<figure>
  <img src="../Images/21.jpg" alt="img21"/>
  <figcaption>
    <h2>Passionate <span>Julia</span></h2>
    <div>
      <p>Julia dances in the deep dark</p>
      <p>She loves the smell of the ocean</p>
      <p>And dives into the morning light</p>
    </div>
    <a href="#fn">View more</a>
  </figcaption>
</figure>
```

发现添加了<figcaption>标签后，无论是语义上还是代码结构上都更加的清晰了。

2.1.9　aside（内容之外的内容）

<aside>标签定义其所处内容之外的内容。aside的内容应该与附近的内容相关。

在ePub 3规范中，aside元素用来组织注释内容。测试如下：

```
<html ... xmlns:epub="http://www.idpf.org/2007/ops">
  ...
  <p> ... <a epub:type="noteref" href="#n1">1</a> ... </p>
  ...
  <aside epub:type="footnote" id="n1">
  ...
  </aside>
  ...
</html>
```

关于aside元素的注释用法，在后面的章节中会着重讲述。

2.1.10　pre（预格式）

pre元素可定义预格式化的文本。被包围在pre元素中的文本通常会保留空格和换行符。而文本也会呈现为等宽字体（指字符宽度相同的电脑字

体，如方块汉字）。

<pre>标签的一个常见应用就是用来表示计算机的源代码。比如在前文中演示p元素时用的诗歌，如果使用pre元素包裹则会显示如下：

```
    春眠不觉晓，
      处处闻啼鸟。
        夜来风雨声，
          花落知多少。
```

测试代码：

```
<pre>
    春眠不觉晓，
      处处闻啼鸟。
        夜来风雨声，
          花落知多少。</pre>
```

可以发现在pre元素中的空格都进行了保留处理，而不会像p元素那样，只留一个半角空格。

> 在用<pre>标签格式化的文档段中使用空格，可以确保文本正确的水平位置。

2.1.11 audio（音频）

<audio>标签定义声音，比如音乐或其他音频流。

> 可以在开始标签和结束标签之间放置文本内容，这样老的浏览器就可以显示出不支持该标签的信息。属性及含义：

- src：音频文件路径。也可以使用<source>标签来设置音频。
- autoplay：设置音频是否自动播放。
- loop：设置音频是否要循环播放。
- controls：显示播放、暂停和音量控件。

测试：

```
<audio src="../Audio/Hit.mp3" controls="controls" autoplay="autoplay" loop="loop">
您的浏览器不支持audio标签。
</audio>
```

显示效果：

　您的浏览器不支持audio标签。

如果你的浏览器不支持audio元素，将显示<audio>和</audio>之间的文本。

使用<source>标签：

```
<audio controls="controls">
  <source src="../Audio/Hit.ogg" type="audio/ogg" />
  <source src="../Audio/Hit.mp3" type="audio/mpeg" />
您的浏览器不支持audio标签。
</audio>
```

对于音频格式，一般提供ogg和mp3格式，就可以支持所有主流浏览器了。

Ogg意指Ogg　Vorbis，是一种新的音频压缩格式，类似于MP3等的音乐格式。Ogg是完全免费、开放和没有专利限制的。Ogg　Vorbis文件的扩展名是.ogg。Ogg文件格式可以不断地进行大小和音质的改良，而不影响旧有的编码器或播放器。

MP3是一种音频压缩技术，其全称是动态影像专家压缩标准音频层面3（Moving Picture Experts Group Audio Layer III），简称为MP3。它被设计用来大幅度地降低音频数据量。利用MPEG　Audio　Layer　3的技术，将音乐以1:10甚至1:12的压缩率，压缩成容量较小的文件，而对于大多数用户来说重放的音质与最初的不压缩音频相比没有明显的下降。

Ogg与MP3在相同位速率（Bit　Rate）编码的情况下，Ogg体积更小，并且Ogg是免费的不用交专利费。

2.1.12　video（视频）

<video>标签定义视频，比如电影片段或其他视频流。

> 可以在<video>开始标签和<video>结束标签之间放置文本内容，这样老的浏览器就可以显示出不支持该标签的信息。

- src：视频文件路径。也可以使用<source>标签来设置音频。
- autoplay：设置视频是否自动播放。
- loop：设置视频是否要循环播放。
- controls：显示播放、暂停和音量控件。

- height：设置视频播放器的高度。
- width：设置视频播放器的宽度。
- poster：规定视频下载时显示的图像或者在用户点击播放按钮前显示的图像。

测试：

```
<video src="../Video/sunup.mp4" controls="controls" poster="../Images/sunup.jpg" height="180">
您的浏览器不支持video标签。
</video>
```

显示效果：

您的浏览器不支持 video 标签。

使用`<source>`标签：

```
<video controls="controls">
  <source src="../Audio/sunup.ogg" type="video/ogg" />
  <source src="../Video/sunup.mp4" type="video/mp4" />
您的浏览器不支持video标签。
</video>
```

MP4，全称MPEG-4 Part 14，是一种使用MPEG-4的多媒体电脑文件格式，扩展名为`.mp4`，以存储数字音频及数字视频为主。MP4至今仍是各大影音分享网站所使用主流格式，MP4优点是压缩质量优、转换容易，目前智能手机录影档，九成以上皆为MP4档。

2.1.13 hr（水平线）

`<hr>`标签在HTML页面中创建一条水平线。水平分隔线可以在视觉上将文档分隔成各个部分。被水平线分隔的段落：

```
<p>这是段落。</p>
<hr />
<p>这是段落。</p>
<hr />
<p>这是段落。</p>
```

显示效果：

这是段落。

这是段落。

这是段落。

在HTML中，<hr>标签没有结束标签。在XHTML中，<hr>必须被正确地关闭，比如<hr />。

2.1.14 nav（导航）

<nav>标签定义导航链接的部分。<nav>标签是HTML 5中的新标签。

> 如果文档中有「前后」按钮，则应该把它放到<nav>标签中。

nav字面上理解为「导航」，在HTML 5中用于包裹一个导航链接组，用于显式说明「这是一个导航组」，在同一个页面中可以同时存在多个nav元素。

1. 不是所有的链接组都需要使用nav包裹，这主要看链接组是不是用于导航（可理解为是不是再页面中充当导航这一角色），比如一本书的目录，它的作用就是起「导航」作用，那么我们就可以用nav元素去组织「目录」内容。
2. 导航，顾名思义，就是引导的路线，那么具有引导功能的都可以认为是导航。不光「目录」内容可以做导航，在一些图书中「表格索引」、「图片索引」、「名词索引」，也可以做「导航」。
3. 导航可以页与页之间导航，也可以是页内的段与段之间导航。
4. nav是一个明确定义功能的元素，所以从文档的意义比功能的意义更大。

如使用ePub 3规范创建的epub文件，「目录」内容即是用nav元素来组织的。部分代码如下：

```
<nav epub:type="toc" id="toc">
  <h1>Table of Contents</h1>
  <ol>
    <li><a href="../Text/Chapter1.xhtml">1 ePub概述</a></li>
    <li><a href="../Text/Chapter2.xhtml">2 HTML知识</a></li>
  </ol>
</nav>
```

以上内容可以在Sigil中的「图书浏览器」窗口中查看*nav.xhtml*文件获取详细代码。

2.2 内联元素

内联元素又名行内元素（inline element），和其对应的是块状元素（block element），都是HTML规范中的概念。内联元素的显示，为了帮助理解，可以形象的称为「文本模式」，即一个挨着一个，都在同一行按从左至右的顺序显示，不单独占一行。特点：

- 和其他元素都在一行上；
- 高，行高及外边距和内边距不可改变；
- 宽度就是它的文字或图片的宽度，不可改变；
- 内联元素只能容纳文本或者其他内联元素。

内联元素通常被包括在块元素中使用，常见内联元素有`<a>`、`<b>`、`<br>`等。

2.2.1 br（换行）

`<br>`可插入一个简单的换行符。换行后的文本使用的是行间距`line-height`，请给`line-height`设置合适的值。

> 请使用`<br>`来输入空行，而不是分割段落（`<p> </p>`）。

在HTML中，`<br>`标签没有结束标签。在XHTML中，`<br>`必须被正确地关闭，比如`<br />`。演示如何在文档中插入换行符：

```
<p>使用<br />标签<br />给<br />句子<br />换行。</p>
```

显示效果：

使用
标签
给

句子
换行。

2.2.2 span（行内组合）

<span>标签被用来组合文档中的行内元素。

> 请使用<span>来组合行内元素，以便通过样式来格式化它们。
> span没有固定的格式表现。当对它应用样式时，它才会产生视觉上的变化。

例子：

```
<p><span>文本。</span>普通文本。</p>
```

例子解释：

如果不对span元素应用样式，那么span元素中的文本与其他文本不会有任何视觉上的差异。尽管如此，上例中的span元素仍然为p元素增加了额外的结构。可以为span应用id或class属性，这样既可以增加适当的语义，又便于对span应用样式。

可以对同一个<span>标签应用class或id属性，但是更常见的情况是只应用其中一种。这两者的主要差异是，class用于元素组（类似的元素，或者可以理解为某一类元素），而id用于标识单独的唯一的元素。比如上面的「提示」内容。

HTML：

```
<p><span class="tips">提示：</span>请使用...</p>
```

CSS：

```
span.tips {
    font-weight: bold; /*字体粗细为加粗*/
    color: #f75555; /*字体颜色*/
}
```

显示效果如上面的「提示」和「注释」。

2.2.3 b（粗体）

该标签传统上被用来表示的意义：关键字、产品名称、引导句……

<b>标签规定粗体文本。在排版中，不建议使用该属性对文本加粗，而是采用经过设计的粗体字体。直接加粗的文本，在显示效果上不如字体设计师设计的字体美观。

> 根据HTML5规范，在没有其他合适标签更合适时，才应该把<b>标签作为最后的选项。HTML 5规范声明：应该使用<h1> ~ <h6>来表示标题，使用<em>标签来表示强调的文本，应该使用<strong>标签来表示重要文本，应该使用<mark>标签来表示标注的、突出显示的文本。
> 您也能够使用CSS`font-weight: bold;`声明来设置粗体文本。

实例：

```
<p>这是普通文本。<b>这是粗体文本</b>。</p>
```

显示效果如下：

这是普通文本。**这是粗体文本**。

<b>标签在Sigil的「工具栏」按钮中可以直接调用，方式：在「书籍视图」或「代码视图」中选择要加粗的文字，点击工具栏按钮中的加粗图标。

2.2.4 i（斜体）

该标签传统上被用来表示的意义：外国文字、分类名称、技术术语、一种思想……

<i>标签显示斜体文本效果。与加粗文本一样，建议使用斜体字体。

<i>标签和基于内容的样式标签<em>类似。它告诉浏览器将包含其中的文本以斜体字（italic）或者倾斜（oblique）字体显示。如果这种斜体字对该浏览器不可用的话，可以使用高亮、反白或加下划线等样式。

> <i>标签一定要和结束标签</i>结合起来使用。您也能够使用`font-style: italic;`声明来设置斜体文本。

实例：

```
<p>这是普通文本。<i>这是斜体文本</i>。</p>
```

显示效果：

这是普通文本。这是斜体文本。

<i>标签在Sigil的「工具栏」按钮中可以直接调用，方式：在「书籍视图」或「代码视图」中选择要加粗的文字，点击工具栏按钮中的斜体图标。

2.2.5 u（下划线）

该标签传统上被用来表示的意义：专有名词、拼写错误……

<u>标签可定义下划线文本。使用<u>标签为文本添加下划线：

```
<p>如果文本不是超链接，就不要<u>对其使用下划线</u>。</p>
```

显示效果：

如果文本不是超链接，就不要对其使用下划线。

> 请尽量避免为文本加下划线——用户会把它混淆为一个超链接！
> 多看不支持<u>这种下划线标签，掌阅则可以。

也可以使用`text-decoration: underline;`声明为选择的文本添加下划线，第3种方式是`border-bottom: 1px solid #333;`，即行内文字加下边框。

关于下划线的更多介绍可以参考解决文字和text-decoration:underline下划线重叠问题这篇文章。

<u>标签在Sigil的「工具栏」按钮中可以直接调用，方式：在「书籍视图」或「代码视图」中选择要加粗的文字，点击工具栏按钮中的下划线图标。

2.2.6 del（被删除）

定义文档中被删除的文本。使用<del>标签为文本添加删除线：

```
<p>今天是2017年7月<del>26</del><ins>27</ins>号。</p>
```

显示效果：

今天是2017年7月2627号。

> 请与<ins>标签配合使用，来描述文档中的更新和修正。

也可以使用`text-decoration: line-thgough;`为选择的文本添加删除线。

<del>标签在Sigil的「工具栏」按钮中可以直接调用，方式：在「书籍视图」或「代码视图」中选择要加粗的文字，点击工具栏按钮中的删除线图标。

2.2.7 sub（下标）

<sub>标签可定义下标文本。

包含在_{标签和其结束标签}中的内容将会以当前文本流中字符高度的一半来显示，但是与当前文本流中文字的字体和字号都是一样的。

> 无论是<sub>标签还是和它对应的<sup>标签，在数学等式、科学符号和化学公式中都非常有用。

示例：

```
<p>这段文本包含<sub>下标</sub>。</p>
<p>水（化学式：H<sub>2</sub>O）是由氢、氧两种元素组成的无机物。</p>
```

显示效果：

这段文本包含下标。

水（化学式：H_2O）是由氢、氧两种元素组成的无机物。

使用`vertical-align: sub;`声明，同时给文字设置字体大小，也可以实现下标的显示效果。

2.2.8 sup（上标）

<sup>标签可定义上标文本。

包含在^{标签和其结束标签}中的内容将会以当前文本流中字符高度的一半来显示，但是与当前文本流中文字的字体和字号都是一样的。

> 这个标签在向文档添加脚注以及表示方程式中的指数值时非常有用。如果和<a>标签结合起

> 来使用，就可以创建出很好的超链接脚注。

示例：

```
<p>这段文本包含<sup>上标</sup>。</p>
<p>勾股定理（表达式：a<sup>2</sup>+b<sup>2</sup>=c<sup>2</sup>）是一个基本的几何定理，指直角三角形的两条直角边的平方和等于斜边的平方。</p>
```

显示效果：

> 这段文本包含上标。
>
> 勾股定理（表达式：$a^2+b^2=c^2$）是一个基本的几何定理，指直角三角形的两条直角边的平方和等于斜边的平方。

使用`vertical-align: super;`声明，同时给文字设置字体大小，也可以实现下标的显示效果。

2.2.9 a（超链接）

`<a>`标签定义超链接，用于从一张页面链接到另一张页面。也可以添加id来在页面内，从一个地方跳转到另一个地方。`<a>`标签最重要的属性是href属性，它指示链接的目标。

> 请使用CSS来设置链接的样式。

你可以在Sigil里使用「工具栏」按钮中的「插入ID」和「插入链接」功能，做一个超链接。具体步骤请见5.3.20节。

创建超链接

本例演示如何在HTML文档中创建链接。

```
<a href="../Text/Chapter2-2.xhtml#test_sup">本文本</a>是一个指向本章中「2.2.8 sup」标题下的正文第一行。</p>
<p><a href="http://www.w3school.com.cn/">本文本</a>是一个指向W3School的首页链接。</p>
```

与之链接的对象代码：

```
<h3><a id="sup-section"></a>2.2.8 sup</h3>
```

显示效果：

本文本是一个指向本章中的「2.2.8 sup」标题下的正文第一行。

本文本是一个指向W3School的首页链接。

在所有浏览器中，链接的默认外观是：

未被访问的链接带有下划线而且是蓝色的

已被访问的链接带有下划线而且是紫色的

活动链接带有下划线而且是红色的

> 您可能已经注意到了，本书内的链接外观与默认的链接外观不同。您可以使用CSS伪类向文本超链接添加复杂而多样的样式。

将图像作为链接

本例演示如何使用图像作为链接。代码：

```
<p>您也可以使用图像来作链接：<a href="../Chapter2-2.xhtml#sup-section"><img alt="next" src="../Images/next.png"/></a></p>
```

显示效果：

您也可以使用图像来作链接：上一节

这个链接将跳转到上一节——2.2.8中。

互联网本身就是靠超链接来连接的，大到整个互联网，小到一个网站都是如此。而ePub电子书，你可以把它看作一个简单的网站，连接这个「网站」的元素是章节标题，当章节标题之间有某种逻辑关系时，就是一本书的目录。

2.2.10 img（图像）

img元素向网页中嵌入一幅图像。

从技术上讲，<img>标签并不会在网页中插入图像，而是从网页上链接图像。<img>标签创建的是被引用图像的占位空间。<img>标签有两个必需的属性：src属性（规定显示图像的路径）和alt属性（规定图像的替代文本，当图像无法显示时）。可选的属性：

- height：定义图像的高度。
- width：设置图像的宽度。

以上两个属性，单位均是像素（pixels）和百分比（%）。

在Sigil里，你可以使用「工具栏」中的「插入文件」按钮来插入一张图像，见5.3.13节。

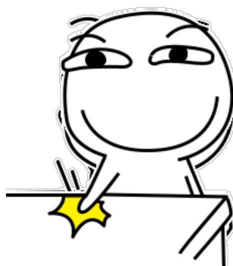
插入图像

本例演示如何在网页中显示图像。

```
<p>一个滑稽图像：<img height="100" alt="滑稽" src="../Images/huaji.jpg"/></p>
<p>一个滑稽动图：<img height="100" alt="滑稽表情" src="../Images/huaji.gif"/></p>
```

显示效果：

一个滑稽图像：

一个滑稽动图：

插入动画图像的语法与插入普通图像的语法没有区别。

> 多看阅读器中GIF格式图像只能显示第1帧，即图像不会「动」。

2.2.11 ruby（行间注）

<ruby>标签定义ruby注释（中文注音或字符）。在东亚使用，显示的是东亚字符的发音。

在zài网页wǎngyè排版páibǎn方面fāngmiàn，由于yóuyú日本Rìběn的de推动tuīdòng，很hěn早zǎo就jiù开始kāishǐ了le对duì注音zhùyīn的de支持zhīchí。早在2001年发布的

IE6浏览器率先就已经部分支持注音，而后W3C发布了「注音」（Ruby Annotation）推荐规范并将其写入HTML规范中，导入<ruby>系列标签进行标注，而排版则可以采用「CSS Ruby 排版模块第一级」的规范内容进行控制。

与<ruby>以及<rt>标签一同使用：ruby元素由一个或多个字符（需要一个解释/发音）和一个提供该信息的rt元素组成，还包括可选的rp元素，定义当浏览器不支持ruby元素时显示的内容。

> 支持ruby元素的浏览器不会显示rp元素的内容。

在使用<ruby>标签制作HTML时应该注意两点：

1. 为实现上述需求「按词拼写还是按字拼写」，应该使用<rb>标签，根据需要把相应的字或者词标注出来。比如按字拼写的「 字 zì 谈 tán 字 zì 畅 chàng 」和按词拼写的「 字谈字畅 zìtánzìchàng 」。
2. 为保证向后兼容，建议用<rp>标签加上括弧，这样可以保证在不支持<ruby>标签的浏览器里，让标注文字从「行间注」回退到带括弧的「行内注」的状态。因此，使用<ruby>的实际代码看起来会比较繁琐，但是，详细的标注意味着在CSS里可以精准地控制，还是值得的。

比如，实现「 孔雀 （ kǒngquè ） 」的实际代码是：

```
<ruby>
    <rb>孔雀</rb>
    <rp>（</rp><rt>kǒngquè</rt><rp>）</rp>
</ruby>
```

一个ruby注释：

```
<ruby>
    <rb>饕</rb>
    <rp>（</rp><rt>tāo</rt><rp>）</rp>
    <rb>餮</rb>
    <rp>（</rp><rt>tiè</rt><rp>）</rp>
</ruby>
<ruby>
    <rb>夏洛克·福尔摩斯</rb>
    <rp>（</rp><rt>Sherlock·Holmes</rt><rp>）</rp>
</ruby>
```

显示效果：

饕 （ tāo ） 餮 （ tiè 　夏洛克·福尔摩斯 （ Sherlock · Holmes ）

需要给ruby元素设置对齐方式，避免rt元素在字符间增减间距以适合ruby元素的宽度。CSS代码：

```
ruby {
    ruby-align: center; /*注音或字符居中*/
}
```

上面的显示效果在不支持`<ruby>`标签的阅读系统中将会显示成这种：饕（tāo）餮（tiè）　夏洛克·福尔摩斯（Sherlock·Holmes）

稳妥的决策指南

拼音字号的大小：不盲目使用软件默认，而是认真测试。绝大多数环境下，正文里的默认设置是按照JIS的「主字的1/2」，而台湾教育部规定注音符号默认约为主字的1/3，这些设置往往都不适合汉语拼音，需要调整。建议拼音字号应该比主字一半再大一些，以55–60%为基准进行调整。

```
ruby rt {
    font-size: .55em; /*注音或字符是父级字号的55%*/
}
```

拼式按「字」还是按「词」：初等教育的材料多是「按字」，而按照正词法应该「按词」。注意这个决策对正文的版面效果影响非常大，需要特别慎重。无论是哪种拼式，都推荐正文要使用中式网格的「疏排」设置，为拼式预留出空间。

拼式的位置：「行内注」按照中西混排用括弧，而「行间注」时最常见的是横排主字之上，偶尔有横排主字之下、竖排主字之右，原则上紧靠主字字框。特别注意主字文章横竖的排版方式、行距的调整对拼音的影响。按照「支持ruby元素的浏览器不会显示rp元素的内容」规则，在`<rp></rp>`标签内加（ ）。

拼式与主字的对齐：原则上选择「居中」，再局部进行调整。自动两端对齐拉伸字间距的做法严重影响阅读，不推荐。如上文居中的方式。

拼式的颜色：是否需要与正文区别处理。

```
ruby rt {
    color: #a52a2a; /*给注音或字符添加颜色*/
}
```

最后，关于拼音字符的字体推荐使用「汉仪新人文宋」或者多看的符号字体（DK-SYMBOL），比如方正仿宋会把有声调符号的字符显示为全

角或显示的注音字符与普通字符不在一个基线上，特别丑。引用外部字体以及调用字体样式：

```
/*汉仪新人文宋*/
@font-face {
    font-family: "汉仪新人文宋"; /*定义字体名称*/
    src: url("../Fonts/HanYiXinRenWenSong.ttf"); /*链接字体资源*/
}
ruby rt {
    font-family: "汉仪新人文宋"; /*注音或字符优先使用「汉仪新人文宋」字体*/
}
```

文章来源：Wǒ ài pīnyīn! 感谢！

2.3 其他元素

本章介绍的是一些其他HTML标签，有些元素我们用不到（软件已经创建好了）或很少使用，还有一些需要了解其意义。

2.3.1 html（网页文档）

此元素可告知浏览器其自身是一个HTML文档。

`<html>`与`</html>`标签限定了文档的开始点和结束点，在它们之间是文档的头部和主体。正如您所了解的那样，文档的头部由`<head>`标签定义，而主体由`<body>`标签定义。

xmlns属性在XHTML中是必需的，但在HTML中不是。不过，即使XHTML文档中的`<html>`没有使用此属性，W3C的验证器也不会报错。这是因为`"xmlns=http://www.w3.org/1999/xhtml"`是一个固定值，即使您没有包含它，此值也会被添加到`<html>`标签中。

> 即使html元素是文档的根元素，它也不包含DOCTYPE元素。DOCTYPE元素必须位于html元素之前。

一个简单的HTML文件：

```
<!DOCTYPE html>
<html>
<head>
  <title>我的第一个HTML页面</title>
</head>
<body>
  <p>body元素的内容会显示在浏览器中。</p>
  <p>title元素的内容会显示在浏览器的标题栏中。</p>
</body>
</html>
```

2.3.2 head（头部）

<head>标签用于定义文档的头部，它是所有头部元素的容器。<head>中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。

文档的头部描述了文档的各种属性和信息，包括文档的标题、在Web中的位置以及和其他文档的关系等。绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

下面这些标签可用在head部分：<base>、<link>、<meta>、<script>、<style>以及<title>。

<title>定义文档的标题，它是head部分中唯一必需的元素。

> 应该把<head>标签放在文档的开始处，紧跟在<html>后面，并处于<body>标签或<frameset>标签之前。
> 请记住始终为文档规定标题！

head元素的位置见上面的案例。

2.3.3 title（文档标题）

<title>元素可定义文档的标题。

浏览器会以特殊的方式来使用标题，并且通常把它放置在浏览器窗口的标题栏或状态栏上。同样，当把文档加入用户的链接列表或者收藏夹或书签列表时，标题将成为该文档链接的默认名称。

我们在制作ePub电子书时，推荐你把当前HTML文件的文章标题放在<title></title>内。如：

```
<head>
  <title>2.3 其他元素</title>
</head>
```

在Sigil中，选择当前HTML文件，点击右键选择「打开方式」，然后选择一个浏览器，打开这个文件。这里使用Chrome浏览器打开，截图如下：



图2-3 浏览器显示<title></title>内的文本

2.3.4 link（外部资源）

<link>标签定义文档与外部资源的关系，最常见的用途是链接样式表。

> link元素是空元素，它仅包含属性。
> 此元素只能存在于head部分，不过它可出现任何次数。

链接到一个外部样式表

本例演示如何<link>标签链接到一个外部样式表。

```
<head>
  <title>2.3 其他元素</title>
  <link href="../Styles/Stylesheet.css" type="text/css" rel="stylesheet"/>
</head>
```

在Sigil中，链接外部样式表只需要在「图书浏览器」中选择要操作的HTML文件，点击鼠标右键，选择「链接样式表」选项。具体步骤见5.3.21节。

2.3.5 style（样式）

<style>标签用于为HTML文档定义样式信息。

在style中，您可以规定在浏览器中如何呈现HTML文档。type属性是必需的，定义style元素的内容。唯一可能的值是"text/css"。style元素位于head部分中。

> 如需链接外部样式表，请使用<link>标签。
> 如需学习更多有关样式表的知识，请阅读本书的下一章。

HTML中的样式

本例演示如何使用添加到<head>部分的样式信息对HTML进行格式化。

```
<html>
<head>
  <title>我的第一个HTML页面</title>
  <style type="text/css">
    h1 { color: red; }
    p { color: blue; }
  </style>
</head>
<body>
  <h1>Chapter1</h1>
  <p>body元素的内容会显示在浏览器中。</p>
</body>
</html>
```

上述代码中，h1元素内的标题会以红色显示，p元素的文本会以蓝色显示。

2.3.6 body（主体）

body元素定义文档的主体。

body元素包含文档的所有内容（比如文本、超链接、图像、表格和列表等等。）一个简单的HTML文件，它向您演示了body元素中的内容是如何被浏览器显示的：

```
<!DOCTYPE html>
<html>
<head>
  <title>我的第一个HTML页面</title>
</head>
<body>
  <p>body元素的内容会显示在浏览器中。</p>
  <p>title元素的内容会显示在浏览器的标题栏中。</p>
</body>
</html>
```

你可以新建一个HTML页面并把body元素中的内容粘贴进去，然后查看显示效果。

2.4 字符实体

HTML中的预留字符必须被替换为字符实体。

在HTML中，某些字符是预留的。在HTML中不能使用小于号（<）和大于号（>），这是因为浏览器会误认为它们是标签。如果希望正确地显示预留字符，我们必须在HTML源代码中使用字符实体（character entities）。字符实体类似这样：

```
&entity_name;
或者
&#entity_number;
```

如需显示小于号，我们必须这样写：`<`或`<`。

> 使用实体名而不是数字的好处是，名称易于记忆。不过坏处是，浏览器也许并不支持所有实体名称（对实体数字的支持却很好）。

HTML中的常用字符实体是不间断空格（` `）。浏览器总是会截短HTML页面中的空格。如果您在文本中写10个空格，在显示该页面之前，浏览器会删除它们中的9个。如需在页面中增加空格的数量，您需要使用` `字符实体。

HTML中有用的字符实体

> 实体名称对大小写敏感！

表3　HTML的字符实体

显示结果	描述	实体名称	实体编号
	空格		
<	小于号	<	<
>	大于号	>	>
&	和号	&	&

> 维基百科上有一个包含所有可用HTML字符实体引用的列表：XML和HTML字符实体引用

列表。

在Sigil中空格的字符实体有常见的4种，它们的含义如下：

它叫不换行空格，全称No-Break Space，它是最常见和我们使用最多的空格，大多数的人可能只接触了 ，它是按下Space键（空格键）产生的空格。在HTML中，如果你用空格键产生此空格，空格是不会累加的（只算1个）。要使用HTML实体表示才可累加，该空格占据宽度受字体影响明显而强烈。

它叫「半角空格」，全称是En Space，en是字体排印学的计量单位，为em宽度的一半。根据定义，它等同于字体宽度的一半（如16px字体中就是8px）。名义上是小写字母n的宽度。此空格传承空格家族一贯的特性：透明的，此空格有个相当稳健的特性，就是其占据的宽度正好是1/2个中文宽度，而且基本上不受字体影响。

它叫「全角空格」，全称是Em Space，em是字体排印学的计量单位，相当于当前指定的点数。例如，1em在16px的字体中就是16px。此空格也传承空格家族一贯的特性：透明的，此空格也有个相当稳健的特性，就是其占据的宽度正好是1个中文宽度，而且基本上不受字体影响。

它叫窄空格，全称是Thin Space。我们不妨称之为「瘦弱空格」，就是该空格长得比较瘦弱，身体单薄，占据的宽度比较小。它是em之六分之一宽。

第3章

CSS知识

层叠样式表（Cascading Style Sheets，简写CSS）是一种用来表现HTML或XML等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化排版。

CSS能够对网页中元素位置的排版进行像素级精确控制，对网页对象和模型样式编辑的能力。借助CSS我们可以完成对图书的排版美化。

CSS是什么，如何深入理解，它和HTML是个怎样的关系？CSS有几种来源？CSS要干什么事情，如何结合HTML实现的？

在解答以上的问题之前，我们先来了解一下浏览器是如何对网页进行渲染的：

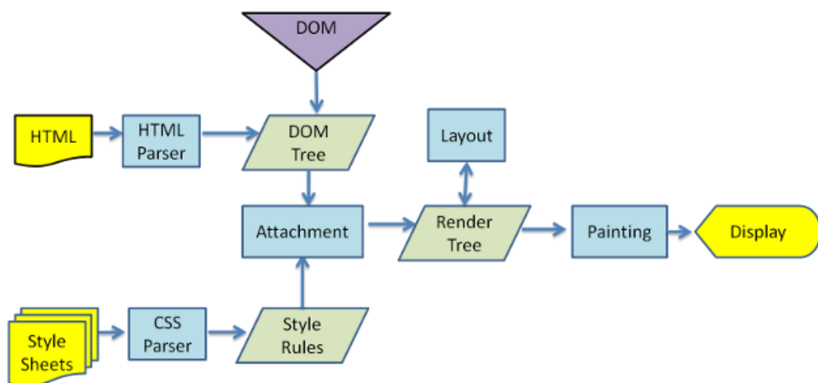


图3-1　webkit内核渲染html和css的流程图

1. 浏览器将从服务器获取的HTML文档构建成文档对象模型DOM（Document Object Model）。
2. 样式将被载入和解析，构成层叠样式表模型CSSOM（CSS Object Model）。
3. 在DOM和CSSOM之上，渲染树（RenderingTree）将会被创建，代表一系列将被渲染的对象。渲染树映射除了不可见元素（例如`<head>`或者含有`display: none;`的标签）外的所有DOM结构。每一段文本字符串都将划分在不同的渲染对象中，每一个渲染对象都包含了它相应的DOM对象以及计算后的样式。换句话讲，渲染树是DOM的直观表示。
4. 渲染树的每个元素包含的内容都是计算过的，它被称之为布局layout。浏览器使用一种流式处理的方法，只需要一次pass绘制操作就可以布局所有的元素（tables需要多次pass绘制，pass表示像素处理和顶点处理）。
5. 最后布局完成，渲染树将转化为屏幕上的实际内容，这一步被称为绘制painting。

可以从图中看到，HTML的解析是一条线，CSS的解析是一条线，两者会在某一点结合，形成最终的视图。

如果我们以CSS为重点看，从上图中我们可以总结出学习CSS的三个突破点。

- 浏览器如何加载和解析CSS——CSS的5个来源；
- CSS和HTML如何结合起来——选择器；
- CSS能控制那些显示方式——盒子模式、浮动、定位、背景、字体等；

CSS的加载过程

CSS——层叠样式表。「样式表」大家比较好理解，那何为「层叠」呢？从字面意思来看，

层叠是需要多层叠加起来。而这个「多层CSS」到底有几层，每一层是什么？这几个问题将是我们讨论的重点。

另外，这么多层次叠加，如果出现冲突怎么处理，以哪个为准？

最后，在这些层次中有一个「浏览器默认样式」层，即浏览器默认的各个HTML元素的样式。

CSS和HTML的结合

CSS是通过选择器与HTML结合。CSS提供了多种多样的选择器类型，而且每个级别都在不断的增加新的选择器类型，使得选择器更加灵活易用。对CSS选择器来说，有一个很重要的话题——级别。

页面呈现

页面呈现可以分为两类——文字，块。

针对文字来说，我们可以设置字体、字号、加粗、斜体、背景色等等；对于块来说情况比较多，有盒子模型、浮动、定位、display、背景等。

本章会重点讲解其中的基础知识以及它们的一些实际应用。

3.1 样式来源

CSS信息可以来自两个方面：

1. 作者样式：作者可以在HTML文件中确定一个外来的、独立的CSS文件（外部样式表），其优先级最低；作者可以将CSS信息包含在HTML文件内（内部样式表）；作者可以在一个HTML元素内结合CSS元素（内联样式），其优先级最高。一般这样做是为了在特殊情况下，把上面来源的CSS覆盖掉；
2. 浏览器样式：假如外部没有特别指定一个样式的话，一般浏览器自己有一个预置的样式。

不同浏览器的默认样式多少有些区别，特别是老版本的浏览器之间，现在高级浏览器越来越向统一的标准靠拢。

CSS之所以有「层叠」的概念，是因为有多个样式来源：内联式、嵌入式、外部式、用户自定义样式、浏览器默认样式。

内联式

在标签的style属性中编写样式。如：

```
<p style="color: red;">p元素内的文本将以红色显示。</p>
```

显示效果：

p元素内的文本将以红色显示。

嵌入式

CSS样式必须写在`<style> ... </style>`之间，并且一般情况下嵌入式CSS样式写在`<head> ... </head>`之间。见2.3.5节。

外部式

CSS样式（也可称为外联式）就是把CSS代码写一个单独的外部文件中，这个CSS样式文件以*.css*为扩展名，在`<head>`内（不是在`<style>`标签内）

使用<link>标签将CSS样式文件链接到HTML文件内。见2.3.4节。

CSS样式文件名称以有意义的英文字母命名，如*main.css*，具体样式名称同样。rel="stylesheet" type="text/css"是固定写法不可修改。<link>标签位置一般写在<head>标签之内。

三种样式的优先级：内联式＞嵌入式＞外联式。但是「嵌入式＞外部式」有一个前提：嵌入式CSS样式的位置一定在外部式的后面（离被设置元素越近优先级别越高）。如：

```
<link href="../Styles/Typesetting.css" type="text/css" rel="stylesheet"/>
<style type="text/css">
  h1 {
      margin: 20% 0 1em;
      border-bottom: 0 solid #333;
      text-align: center;
  }
  p {
      text-indent: 2em;
  }
</style>
```

用户自定义样式表

在一些网站中，经常看到可以设置字体大小的快捷菜单，例如下图就是起点中文网中的设置：



图3-2　起点中文网的用户自定义样式

用户在这里设置了字体和字号之后，它们会覆盖掉浏览器默认样式和自

定义样式。

浏览器默认样式表

当你不为HTML设置任何样式时，显示在浏览器上，`<b>`标签会显示粗体、`<p>`有纵向margin、`<h1>`字号比`<p>`大一倍……这是为什么呢？

因为浏览器自带一个默认的样式，如果HTML中没有为标签设置样式，则浏览器会按照自己的样式来显示。但是浏览器默认样式的级别是最低的，一旦有其他地方设置了标签样式，浏览器默认样式就会被覆盖掉。不同浏览器的默认样式有些地方是不一样的。

例如当前HTML文件用Chrome浏览器打开，然后使用开发者工具（F12）查看当前段落的属性，下图即是p元素的默认样式：

```
Filter                :hov  .cls  +
element.style {
}
p {                 user agent stylesheet
   display: block;
   -webkit-margin-before: 1em;
   -webkit-margin-after: 1em;
   -webkit-margin-start: 0px;
   -webkit-margin-end: 0px;
}
```

display	block
dominant-baseline	auto
empty-cells	show
fill	rgb(0, 0, 0)
fill-opacity	1
fill-rule	nonzero
filter	none
flex-basis	auto
flex-direction	row
flex-grow	0
flex-shrink	1
flex-wrap	nowrap
float	none
flood-color	rgb(0, 0, 0)
flood-opacity	1
font-family	"Microsoft YaHei"
font-feature-settings	normal
font-kerning	auto
font-size	16px
font-stretch	normal
font-style	normal

图3-3　Chrome浏览器中p元素的默认样式（部分）

3.2 层叠规则

CSS（层叠样式表）是浏览器对多个样式来源进行叠加，最终确定结果的过程。

由于样式的来源不同，浏览器在加载样式时，需要计算出最终的样式值，这样才能显示出正确的界面效果——浏览器会通过叠加和覆盖这两种方式来生成最终的样式值。

CSS样式优先级按照就近原则（离被设置元素越近优先级别越高）确定。

下面是举例说明，如下面代码。第一个引自外部式、第二个嵌入式：

```
p {
    color: red; /*字体颜色为红色*/
    font-weight: bold; /*字体粗细为加粗*/
} /*来自样式表文件*/
p {
    color: blue; /*字体颜色为蓝色*/
} /*来自<style>标签*/
```

不同来源的两个样式，第一个样式设置了`font-weight`，第二个没有，浏览器会把它叠加在一起，即浏览器会把各个零散的整合成一个整体。第一个样式`color: red;`，第二个样式`color: blue;`，根据就近原则，浏览器会让后者覆盖前者，最终结果还是`color: blue;`。最后显示出来的代码：

```
p {
    color: blue; /*字体颜色为蓝色*/
    font-weight: bold; /*字体粗细为加粗*/
}
```

覆盖的默认规则是后者覆盖前者，但是有一个特殊情况——`!important`。如下面代码：

```
span {
    color: red !important; /*字体颜色优先显示为红色*/
} /*来自样式表*/
span {
    color: blue;
} /*来自<style>标签*/
```

最终显示出来的代码如下：

```
span {
    color: red; /*字体颜色显示为红色*/
}
```

虽然`color: blue;`是后来者，但是它没有居上，因为前者`color: red;`跟着`!important`。因为`!important`的作用是提高指定CSS样式规则的应用优先权。

3.3 选择器

在CSS中，选择器是一种模式，用于选择需要添加样式的元素。

CSS由多组「规则」组成。每个规则由「选择器」（selector）、「属性」（property）和「值」（value）组成：

1. 选择器（Selector）：多个选择器可以半角逗号（,）隔开。
2. 属性（property）：CSS1、CSS2、CSS3规定了许多的属性，目的在控制选择器的样式。
3. 值（value）：指属性接受的设置值，多个关键字时大都以空格隔开。

属性和值之间用半角冒号（:）隔开，属性和值合称为「特性」。多个特性间用「;」隔开，最后用「{}」括起来。如下图所示：



图3-4　一个样式的构成

要针对没有特定语义的标签进行样式设置时，可利用`<div>`与`<span>`标签包裹内容，CSS里现在共有5种基本选择器（Basic Selectors）和2种伪选择器。

3.3.1 通用选择

该选择器可以与任何元素匹配，就像是一个通配符。例如，下面的规则可以使文档中的每个元素都为红色：

```
* {
```

```
    color: red;
}
```

* 选择器选取所有元素。

* 选择器也能选取另一个元素中的所有元素。示例：

选取`<div>`元素内部的所有元素：

```
div * {
    background-color: yellow; /*背景颜色显示为黄色*/
}
```

3.3.2　单标签

单标签选择器又叫元素选择器，选择器通常是某个HTML元素，比如`<p>`、`<h1>`、`<em>`、`<a>`，甚至可以是`<html>`本身：

```
html {
    color: black;
}
h1 {
    color: blue;
}
h2 {
    color: silver; /*<h2>标签的内容以银色显示*/
}
```

元素选择器用于指定元素名称的所有元素。

3.3.3　多标签

多标签选择器一般和HTML上下文有关，它有以下几种分类：

1. 选择一个祖先的所有子孙节点，例如div p{...}
2. 选择一个父元素的所有直属节点，例如div > p{...}
3. 选择某一个元素紧挨着的兄弟节点，例如li + li{...}
4. 选择某一个元素的所有同胞节点，例如span ~ a{...}
5. 以上各种情况的组合应用（不要组合过于复杂，编码讲求可读性第一）

给大家列举一个比较典型的应用：

- 列表1
- 列表2
- 列表3

- 列表4
- 列表5

代码：

```
<ul>
  <li style="border-bottom: 1px solid #d2d2d2;">列表1</li>
  <li style="border-bottom: 1px solid #d2d2d2;">列表2</li>
  <li style="border-bottom: 1px solid #d2d2d2;">列表3</li>
  <li style="border-bottom: 1px solid #d2d2d2;">列表4</li>
  <li>列表5</li>
</ul>
```

上面中的效果应该比较常见，在各个列表之间加下划线。通常实现的方法是：每个`<li>`都加一个`border-bottom`，再把最后一个`<li>`的`border-bottom`去掉。

其实完全没必要这样麻烦，下面一个样式设置即可解决：

```
ul li+li {
    border-top: 1px solid #d2d2d2;
} /*相邻的<li>标签，第2个<li>标签显示上边框*/
```

3.3.4　id选择器

在某些方面，ID选择器类似于类选择器，不过也有一些重要差别。

语法

首先，ID选择器前面有一个#号——也称为棋盘号或井号。请看下面的规则：

```
*#intro {
    font-weight: bold;
}
```

与类选择器一样，ID选择器中可以忽略通配选择器。前面的例子也可以写作：

```
#intro {
    font-weight: bold;
}
```

这个选择器的效果将是一样的。

第二个区别是ID选择器不引用class属性的值，毫无疑问，它要引用id属性中的值。

以下是一个实际ID选择器的例子：

```
<p id="intro">This is a paragraph of introduction.</p>
```

类选择器还是ID选择器？

在类选择器这一节中讲解过，可以为任意多个元素指定类。下一节中类名important被应用到p和h1元素，而且它还可以应用到更多元素。

区别1：只能在文档中使用一次

与类不同，在一个HTML文档中，ID选择器会使用一次，而且仅一次。

区别2：不能使用ID词列表

不同于类选择器，ID选择器不能结合使用，因为id属性不允许有以空格分隔的词列表。

区别3：ID能包含更多含义

类似于类，可以独立于元素来选择id。有些情况下，您知道文档中会出现某个特定id值，但是并不知道它会出现在哪个元素上，所以您想声明独立的ID选择器。例如，您可能知道在一个给定的文档中会有一个id值为mostImportant的元素。您不知道这个最重要的东西是一个段落、一个短语、一个列表项还是一个小节标题。您只知道每个文档都会有这么一个最重要的内容，它可能在任何元素中，而且只能出现一个。在这种情况下，可以编写如下规则：

```
#mostImportant {
    color: red;
    background: yellow;
}
```

这个规则会与以下各个元素匹配（这些元素不能在同一个文档中同时出现，因为它们都有相同的id值）：

```
<h1 id="mostImportant">This is important!</h1>
<em id="mostImportant">This is important!</em>
<ul id="mostImportant">This is important!</ul>
```

区分大小写

请注意，类选择器和ID选择器可能是区分大小写的。这取决于文档的语

言。HTML和XHTML将类和id值定义为区分大小写，所以类和id值的大小写必须与文档中的相应值匹配。因此，对于以下的CSS和HTML，元素不会变成粗体：

```
#intro {
    font-weight: bold;
}
<p id="Intro">This is a paragraph of introduction.</p>
```

3.3.5　类选择器

类选择器允许以一种独立于文档元素的方式来指定样式。该选择器可以单独使用，也可以与其他元素结合使用。

> 只有适当地标记文档后，才能使用这些选择器，所以使用这两种选择器通常需要先做一些构想和计划。

修改HTML代码

在使用类选择器之前，需要修改具体的文档标记，以便类选择器正常工作。

为了将类选择器的样式与元素关联，必须将class指定为一个适当的值。请看下面的HTML代码：

```
<h1 class="important">This heading is very important.</h1>
<p class="important">This paragraph is very important.</p>
```

在上面的代码中，两个元素的class都指定为important：第一个标题（h1元素），第二个段落（p元素）。

语法

然后我们使用以下语法向这些归类的元素应用样式，即类名前有一个点号（.），然后结合通配选择器：

```
*.important {
    color: red;
}
```

如果您只想选择所有类名相同的元素，可以在类选择器中忽略通配选择器，这没有任何不好的影响：

```
.important {
```

```
    color: red;
}
```

结合元素选择器

类选择器可以结合元素选择器来使用。例如，您可能希望只有段落显示为红色文本：

```
p.important {
    color: red;
}
```

选择器现在会匹配class属性包含important的所有p元素，但是其他任何类型的元素都不匹配，不论是否有此class属性。选择器`p.important`解释为：「其class属性值为important的所有段落」。因为h1元素不是段落，这个规则的选择器与之不匹配，因此h1元素不会变成红色文本。

如果你确实希望为h1元素指定不同的样式，可以使用选择器`h1.important`：

```
p.important {
    color: red;
}
h1.important {
    color: blue;
}
```

CSS 多类选择器

在HTML中，一个class值中可能包含一个词列表，各个词之间用空格分隔。例如，如果希望将一个特定的元素同时标记为重要（important）和警告（warning），就可以写作：

```
<p class="important warning">This paragraph is a very important warning.</p>
```

这两个词的顺序无关紧要，写成`warning important`也可以。

我们假设class为important的所有元素都是粗体，而`class="warning"`的所有元素为斜体，class中同时包含important和warning的所有元素还有一个银色（silver）的背景 。就可以写作：

```
.important {
    font-weight: bold;
}
.warning {
```

```
    font-style: italic;
}
.important.warning {
    background: silver;
}
```

通过把两个类选择器链接在一起，仅可以选择同时包含这些类名的元素（类名的顺序不限）。

如果一个多类选择器包含类名列表中没有的一个类名，匹配就会失败。请看下面的规则：

```
.important.urgent {
    background: silver;
}
```

不出所料，这个选择器将只匹配class属性中包含词important和urgent的p元素。因此，如果一个p元素的class属性中只有词important和warning，将不能匹配。不过，它能匹配以下元素：

```
<p class="important urgent warning">This paragraph is a very important and urgent warning.</p>
```

3.3.6　属性选择器

属性选择器有两种情况：

1. 只通过属性名选择：img[title]{...}
2. 通过属性名和属性值选择：input[type="text"]{...}

如果希望选择有某个属性的元素，而不论属性值是什么，可以使用简单属性选择器。如果把包含标题（title）的所有元素变为红色，可以写作：

```
*[title] {
    color: red;
}
```

可以只对有href属性的锚（a元素）应用样式：

```
a[href] {
    color: red;
}
```

还可以根据多个属性进行选择，只需将属性选择器链接在一起即可。为

了将同时有href和title属性的HTML超链接的文本设置为红色，可以这样写：

```
a[href][title] {
    color: red;
}
```

这种选择器类似正则表达式，更多详细介绍，可参考W3school。

3.3.7 伪类

伪类分为UI伪类和结构化伪类。UI伪类都比较简单常用：

```
a:link {
    color: #ff0000;
} /*未访问的链接*/
a:visited {
    color: #00ff00;
} /*已访问的链接*/
a:hover {
    color: #ff00ff;
} /*鼠标移动到链接上*/
a:active {
    color: #0000ff;
} /*选定的链接*/
```

为了便于后期维护和可以明确表示每个样式的含义，建议对选择器添加注释。注释格式如上：注释内容放在`/*`和`*/`之间。

结构化伪类，比如如何实现一个表格间隔显示背景颜色？如下所示：

Trial	Business	Team	Solo
15	35	15	5
1GB	5GB	2GB	500MB
√	√	√	---
Unlimited	Unlimited	Unlimited	Unlimited

表格HTML代码：

```
<table class="table_demo">
  <tbody>
    <tr>
      <th>Trial</th>
      <th>Business</th>
      <th>Team</th>
      <th>Solo</th>
    </tr>
```

```html
    <tr>
      <td>15</td>
      <td>35</td>
      <td>15</td>
      <td>5</td>
    </tr>
    <tr>
      <td>1GB</td>
      <td>5GB</td>
      <td>2GB</td>
      <td>500MB</td>
    </tr>
    <tr>
      <td>√</td>
      <td>√</td>
      <td>√</td>
      <td>---</td>
    </tr>
    <tr>
      <td>Unlimited</td>
      <td>Unlimited</td>
      <td>Unlimited</td>
      <td>Unlimited</td>
    </tr>
  </tbody>
</table>
```

表格CSS代码如下：

```css
<style type="text/css">
  table.table_demo {
      margin: .75em auto;
      border: 1px solid #75715e;
      border-collapse: collapse;
  } /*定义表格样式名称并合并table和td元素的边框线*/
  table.table_demo td, th {
      padding: .25em;
      border: 1px solid #75715e;
      text-align: center;
      vertical-align: middle;
      font-family: "黑体";
  } /*定义td和th元素的样式，如填充、水平垂直居中等*/
  table.table_demo th {
      background-color: #272822;
      color: #fafafa;
  } /*设置表头的背景颜色和字体颜色*/
  table.table_demo td {
```

```
        font-size: .85em;
    }
    table.table_demo th:nth-child(1) {
        background-color: #256300;
    } /*设置表头的第一个单元格背景颜色，即绿色*/
    table.table_demo td:nth-child(1) {
        background-color: #ddeedb;
    } /*设置第一列单元格背景颜色*/
    table.table_demo tr:nth-child(odd) {
        background-color: #e1e1e1;
    } /*设置横向偶数行单元格背景颜色*/
</style>
```

最简单的方式莫过于使用结构化伪类，一句样式设置即可实现。而不需要添加class或id来定义每个tr、th、td元素。odd表示偶数，even表示奇数。

如果上面的代码你现在还看不懂，不要着急，在3.4中会讲。

关于结构化伪类可以参考Family.scss网站，不过在多看阅读器中结构化伪类并不支持，所以如果美化表格时还需要添加类。

3.3.8 伪元素

:first-line伪元素

用于向文本的首行设置特殊样式。

> 「first-line」伪元素只能用于块级元素。

如：

```
<p class="firsr-line">王菲（英文名：Faye Wong，1969年8月8日 - ），曾用艺名王靖雯（英文名：Shirley Wong），著名华语女歌手、演员，出生于中国北京，1987年随父亲举家移民英属香港，并于1989年以专辑《王靖雯》出道。王菲在华人世界拥有极高知名度，是近二十年内华语乐坛最出色的歌手之一，被誉为大中华地区的「乐坛天后」。</p>
```

CSS代码：

```
p.firsr-line:first-line {
    font-family: "楷体";
    color: red;
} /*第1行以红色楷体字显示*/
```

显示效果：

王菲（英文名：Faye Wong，1969年8月8日 - ），曾用艺名王靖雯（英文名：Shirley　Wong），著名华语女歌手、演员，出生于中国北京，1987年随父亲举家移民英属香港，并于1989年以专辑《王靖雯》出道。王菲在华人世界拥有极高知名度，是近二十年内华语乐坛最出色的歌手之一，被誉为大中华地区的「乐坛天后」。

如果你是使用多看阅读器浏览上面的例子，是看不到任何变化的，原因是多看并没有支持该属性。

:first-letter伪元素

用于向文本的首字母设置特殊样式。

> 「first-letter」伪元素只能用于块级元素。

仍然使用上面的测试代码，在CSS中在添加一个样式：

```
p.firsr-line:first-letter {
    font-family: "黑体";
    font-size: 1.25em;
    color: blue;
} /*第1行第1个字符以蓝色、1.25倍父级字号、黑体字显示*/
```

显示效果如上所示。这个伪元素可以帮助我们在排电子书时对每段的第一个字符做设计，比如首字下沉效果。

:before伪元素

可以在元素的内容前面插入新内容。

:after伪元素

可以在元素的内容之后插入新内容。可以用于清除浮动的样式。

> 请使用content属性来指定要插入的内容。

我们在做标题设计时，需要在标题前或标题后添加特殊的修饰字符或icon图标，就可以使用上面的两个伪元素。比如在文章中随处可见的棕色小标题，我想在每个小标题前添加一个实心方块■。则可以这么写：

```
p.before-after:before {
    content: "■";
    margin-right: .25em;
```

```
} /*在元素之前添加内容■，距元素0.25倍父级字号*/
```

再为小标题后添加一个图标：

```
p.before-after:after {
    content: url(../Images/title.png);
    margin-left: .25em;
} /*在元素之后添加图标，距元素0.25倍父级字号*/
```

测试：

伪元素测试

它的优点在于不用在每个小标题中增加■字符，这对于管理来说，只要删除CSS中这个属性，就可以在全文中移除这个样式。

如果你看不到任何效果，可能是你的阅读器并不支持该属性。

3.4 属性和值

在上一章介绍了选择器的概念，其中就使用了不少属性来演示，这一章就专门介绍CSS属性和值的。然后在电子书制作中，我习惯称呼一个「选择器＋声明」为一个样式，不同的样式组合成一个大的样式。比如书信样式，它就是由3种小样式组成：书信的称谓、书信的正文、书信的落款。这3中样式组合在一起使用，就构成了一个「书信」样式。

样式的定义：指用有意义的名称保存的字符格式和段落格式的集合。

而每个样式就是由不同的属性组成的，它们是样式的基础单位。下面分别介绍每个属性的用法，并附有实际案例。

关于多看阅读/掌阅与CSS的兼容性，可以参考 @云轩阁主 写的文章。

3.4.1 动画属性

多看阅读和掌阅都不支持动画效果，请使用Gitden和Lithium阅读器查看动画运行效果。

@keyframes规则

通过`@keyframes`规则，您能够创建动画。创建动画的原理是，将一套CSS样式逐渐变化为另一套样式。在动画过程中，您能够多次改变这套CSS样式。以百分比来规定改变发生的时间，或者通过关键词from和to，等价于0%和100%。0%是动画的开始时间，100%动画的结束时间。为了获得最佳的浏览器支持，您应该始终定义0%和100%选择器。

> 请使用动画属性来控制动画的外观，同时将动画与选择器绑定。

语法：

```
@keyframes animationname {
  keyframes-selector {
    css-styles;
  }
}
```

解释：

- animationname：必需。定义动画的名称。
- keyframes-selector：必需。动画时长的百分比。0-100%（from与0%相同，to与100%相同）
- css-styles：必需。一个或多个合法的CSS样式属性。

animation属性

`animation`属性是一个简写属性，用于设置六个动画属性：

1. animation-name：规定需要绑定到选择器的keyframe名称。
2. animation-duration：规定完成动画所花费的时间，以秒或毫秒计。
3. animation-timing-function：规定动画的速度曲线。
4. animation-delay：规定在动画开始之前的延迟。
5. animation-iteration-count：规定动画应该播放的次数。
6. animation-direction：规定是否应该轮流反向播放动画。

语法：

```
animation: name duration timing-function delay iteration-count direction;
```

「旋转太极」案例：

[redacted]

Sigil中不能显示动画效果，想要查看的话，请使用Chrome浏览器打开当前HTML文件。

HTML代码：

```
<div class="animation" id="one">
  <div id="two"> </div>
  <div id="three">
    <div id="five"> </div>
  </div>
  <div id="four">
    <div id="six"> </div>
  </div>
</div>
```

CSS代码：

```
<style type="text/css">
  @keyframes mymove {
      from { transform: rotateZ(360deg); }
      to { transform: rotateZ(0deg); }
  } /*定义动画样式在Z轴上旋转*/
  .animation {
      animation-name: mymove; /*定义动画名称mymove*/
      animation-duration: 5s; /*动画运行时间*/
      animation-timing-function: linear; /*动画的速度曲线*/
      animation-iteration-count: infinite; /*动画的播放次数为无限*/
      -webkit-animation-name: mymove;
      -webkit-animation-duration: 5s;
      -webkit-animation-timing-function: linear;
      -webkit-animation-iteration-count: infinite;
  } /*定义动画属性*/
  #one {
      width: 200px;
      height: 200px;
      background-color: #000;
      border-radius: 50%;
      box-shadow: 0 0 70px #e1e1e1;
      margin: 1.5em auto 0 auto;
  } /*底部黑色大圆*/
```

```
    #two {
        background-color: #fff;
        width: 200px;
        height: 100px;
        border-top-left-radius: 100px;
        border-top-right-radius: 100px;
    } /*上部白色半圆*/
    #three {
        background-color: #fff;
        width: 100px;
        height: 100px;
        border-radius: 50%;
        margin-top: -50px;
        position: relative;
    } /*左侧白色小圆*/
    #four {
        background-color: #000;
        width: 100px;
        height: 100px;
        border-radius: 50%;
        margin-top: -100px;
        margin-left: 100px;
        position: relative;
    } /*右侧黑色小圆*/
    #five {
        background-color: #000;
        width: 20px;
        height: 20px;
        border-radius: 50%;
        position: absolute;
        top: 40px;
        left: 40px;
    } /*左侧白色小圆中的黑色圆心*/
    #six {
        background-color: #fff;
        width: 20px;
        height: 20px;
        border-radius: 50%;
        position: absolute;
        top: 40px;
        left: 40px;
    } /*右侧黑色小圆中的白色圆心*/
</style>
```

animation-name属性

为`@keyframes`动画规定名称。语法：

```
animation-name: keyframename | none;
```

解释：

- keyframename：规定需要绑定到选择器的keyframe的名称。
- none：规定无动画效果（可用于覆盖来自级联的动画）。

当前属性的值为mymove。

animation-duration属性

定义动画完成一个周期所需要的时间，以秒或毫秒计。语法：

```
animation-duration: time;
```

解释：

- time：规定完成动画所花费的时间。默认值是0，意味着没有动画效果。

当前属性的值是5s。

animation-timing-function属性

规定动画的速度曲线。速度曲线定义动画从一套CSS样式变为另一套所用的时间。速度曲线用于使变化更为平滑。语法：

```
animation-timing-function: value;
```

`animation-timing-function`使用名为三次贝塞尔（Cubic Bezier）函数的数学函数，来生成速度曲线。您能够在该函数中使用自己的值，也可以使用预定义的值。

- linear：动画从头到尾的速度是相同的。
- ease：默认。动画以低速开始，然后加快，在结束前变慢。
- ease-in：动画以低速开始。
- ease-out：动画以低速结束。
- ease-in-out：动画以低速开始和结束。
- cubic-bezier(n,n,n,n)：在cubic-bezier函数中输入自己的值。可能的值是从0到1的数值。

当前属性的值是linear。

animation-iteration-count属性

定义动画的播放次数。语法：

```
animation-iteration-count: n | infinite;
```

解释：

- *n*：定义动画播放次数的数值。
- infinite：规定动画应该无限次播放。

当期属性的值是infinite。

animation-delay属性

定义动画何时开始。值以秒或毫秒计。

> 允许负值，-2s使动画马上开始，但跳过2秒进入动画。

语法：

```
animation-delay: time;
```

解释：

- time：可选。定义动画开始前等待的时间，以秒或毫秒计。默认值是0。

当前属性未设置。

animation-direction属性

定义是否应该轮流反向播放动画。

> 如果把动画设置为只播放一次，则该属性没有效果。

语法：

```
animation-direction: normal | alternate;
```

解释：

- normal：默认值。动画应该正常播放。
- alternate：动画应该轮流反向播放。

当前属性未设置。

3.4.2 背景属性

`background`简写属性在一个声明中设置所有的背景属性。可以设置如下属性：

background-color属性

规定要使用的背景颜色。这个属性接受任何合法的颜色值。

`background-color`属性为元素设置一种纯色。这种颜色会填充元素的内容、内边距和边框区域，扩展到元素边框的外边界（但不包括外边距）。如果边框有透明部分（如虚线边框），会透过这些透明部分显示出背景色。

尽管在大多数情况下，没有必要使用transparent。不过如果您不希望某元素拥有背景色，同时又不希望用户对浏览器的颜色设置影响到您的设计，那么设置transparent值还是有必要的。

本例演示如何为元素设置背景颜色：

```
<p>background-color不能继承，其默认值是transparent。transparent有「<span class="highlight">透明</span>」之意。也就是说，如果一个元素没有指定背景色，那么背景就是透明的，这样其祖先元素的背景才能可见。</p>
```

显示效果：

> background-color不能继承，其默认值是transparent。transparent有「透明」之意。也就是说，如果一个元素没有指定背景色，那么背景就是透明的，这样其祖先元素的背景才能可见。

选择器highlight的CSS代码如下：

```
span.highlight {
    background-color: yellow;
}
```

也可以为一个块元素添加背景颜色，如果您希望背景色从元素中的文本向外少有延伸，只需增加一些内边距：

```
<p style="background-color: #e1e1e1;padding: 10px;">background-color不能继承，其默认值是transparent。transparent有「<span class="highlight">透明</span>」之意。也就是说，如果一个元素没有指定背景色，那么背景就是透明的，
```

```
这样其祖先元素的背景才能可见。</p>
```

显示效果：

> background-color不能继承，其默认值是transparent。transparent有「透明」之意。也就是说，如果一个元素没有指定背景色，那么背景就是透明的，这样其祖先元素的背景才能可见。

可能的值：

- color_name：规定颜色值为颜色名称的背景颜色（比如red）。
- hex_number：规定颜色值为十六进制值的背景颜色（比如#ff0000）。
- rgb_number：规定颜色值为 rgb 代码的背景颜色（比如rgb(255,0,0)）。
- transparent：默认。背景颜色为透明。
- inherit：规定应该从父元素继承background-color属性的设置。

还有一种值的写法是rgba，即增加alpha（透明度），其中r指的是red（红色值）、g指的是green（绿色值）、b指的是blue（蓝色值）。如：

光的三原色

在Sigil编辑界面可以垂直居中，但是在多看阅读器中是垂直顶部。

HTML代码：

```
<div class="light-3">
  <p>光的三原色</p>
</div>
```

CSS代码：

```
div.light-3 {
    margin: .65em auto;
    width: 8em;
    height: 4.95em;
    background-color: rgba(255,0,255,0.2);
}
div.light-3 p {
    margin: 0;
    text-align: center;
    line-height: 4.95em; /*行高为容器的高度，即可垂直居中*/
```

```
    color: rgb(255,0,255);
}
```

其中(255,0,255)的显示颜色是品红，alpha取值范围是0到1，这里设置的是0.2，即透明度20%，值越大透明度越低。

background-image属性

为元素设置背景图像。

元素的背景占据了元素的全部尺寸，包括内边距和边框，但不包括外边距。默认地，背景图像位于元素的左上角，并在水平和垂直方向上重复。

> 请设置一种可用的背景颜色，这样的话，假如背景图像不可用，页面也可获得良好的视觉效果。

可能的值：

- url("URL")：指向图像的路径。
- none：默认值。不显示背景图像。
- inherit：规定应该从父元素继承background-image属性的设置。

演示如何将图像设置为背景：

HTML代码：

```
<div class="background-img-demo"> </div>
```

CSS代码：

```
div.background-img-demo {
    margin: .65em auto;
    width: 380px;
    height: 220px;
    background-image: url("../Images/cookies.png");
    background-repeat: repeat;
    border: 1px solid #777;
} /*margin左右设置auto属性的效果即是水平居中*/
```

也可以为body元素应用背景颜色或图像，这样就可以为一本书定义背景图案。前提是你给body元素添加了样式。

background-size属性

规定背景图像的尺寸。语法：

```
background-size: length | percentage | cover | contain;
```

可能的值：

- *length*：设置背景图像的高度和宽度。第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会被设置为"auto"。
- percentage：以父元素的百分比来设置背景图像的宽度和高度。第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会被设置为"auto"。
- cover：把背景图像扩展至足够大，以使背景图像完全覆盖背景区域。背景图像的某些部分也许无法完全显示在背景区域中。
- contain：把图像图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域。

下面分别给背景设置不同的值，HTML的代码只贴出一个，CSS代码一样，不同的值在每个案例中只修改对应的。XXX即要设置的值。

HTML代码：

```
<div class="background-img-XXX"> </div>
```

CSS代码：

```
div.background-img-XXX {
```

```
    margin: .65em auto;
    width: 300px;
    height: 220px;
    background-image: url("../Images/huaji.jpg");
    background-size: XXX;
    border: 1px solid #777;
}
```

首先是length，这个值在上述案例中已经演示过了，这里不再贴出。然后是`percentage: 100% 100%;`，会拉伸图片以适合容器宽度：



cover值（图片适合容器宽度，但是可能下方部分无法显示）：



contain值（增加了背景重复和颜色属性；图片适合容器高度）：

从上面的例子中，可以看到，当我们需要一张图片完全显示在一个容器中，就可以设置`percentage: 100% 100%;`即可。

background-repeat属性

设置是否及如何重复背景图像。

默认地，背景图像在水平和垂直方向上重复。定义了图像的平铺模式。

> 背景图像的位置是根据`background-position`属性设置的。如果未规定`background-position`属性，图像会被放置在元素的左上角。

可能的值：

- repeat：默认。背景图像将在垂直方向和水平方向重复。
- repeat-x：背景图像将在水平方向重复。
- repeat-y：背景图像将在垂直方向重复。
- no-repeat：背景图像将仅显示一次。
- inherit：规定应该从父元素继承background-repeat属性的设置。

本例演示如何重复背景图像。

如何在垂直方向重复背景图像：

如何在水平方向重复背景图像：

如何仅显示一次背景图像：

为了可以在分屏阅读器中完整显示，我给每个案例添加强制分页属性。

background-position属性

设置背景图像的起始位置。

这个属性设置背景原图像（由`background-image`定义）的位置，背景图像如果要重复，将从这一点开始。

> 您需要设置`background-attachment: fixed;`，才能保证该属性在Firefox和Opera中正常工作。

可能的值：

- top left|top center|top right|center left|center center|center right|bottom left|bottom center|bottom right：如果您仅规定了一个关键词，那么第二个值将是"center"。默认值：0% 0%。
- x% y%：第一个值是水平位置，第二个值是垂直位置。左上角是0% 0%。右下角是100% 100%。如果您仅规定了一个值，另一个值将是50%。
- xpos ypos：第一个值是水平位置，第二个值是垂直位置。左上角是0 0。单位是像素(0px 0px) 或任何其他的CSS单位。如果您仅规定了一个值，另一个值将50%。您可以混合使用% 和position值。

同样用上面的例子做演示：

HTML代码：

```
<div class="background-img-position"> </div>
```

CSS代码：

```
div.background-img-position {
    margin: .65em auto;
    width: 380px;
    height: 220px;
    background-image: url("../Images/cookies.png");
    border: 1px solid #777;
    background-repeat: no-repeat;
    background-position: center center;
}
```

其他值可以自己更改，查看实际显示效果。

background-clip属性

规定背景的绘制区域。语法：

```
background-clip: border-box | padding-box | content-box;
```

HTML代码（以下演示只更改了样式名，结构未变）：

```
<div class="bgclip-border"> </div>
```

解释：

- border-box：背景填充到边框中。

显示效果：



CSS代码：

```
div.bgclip-border {
    width: 220px;
    height: 120px;
    background-color: #f75555;
    background-clip: border-box;
```

```
    border: 10px dotted #e1e1e1;
}
```

解释：

- padding-box：背景填充到内边距框。

显示效果：



CSS代码：

```
div.bgclip-padding {
    width: 200px;
    height: 100px;
    padding: 10px;
    background-color: #f75555;
    background-clip: padding-box;
    border: 10px dotted #ff00ff;
}
```

解释：

- content-box：背景填充到内容框。

显示效果：



CSS代码：

```
div.bgclip-content {
    width: 200px;
    height: 100px;
    padding: 10px;
    background-color: #f75555;
    background-clip: content-box;
    border: 10px dotted #ff00ff;
}
```

background-origin属性

规定`background-position`属性相对于什么位置来定位。

用法和上面的`background-clip`属性一样。语法：

```
background-origin: padding-box | border-box | content-box;
```

background-attachment属性

设置背景图像是否固定或者随着页面的其余部分滚动。

可能的值：

- scroll：默认值。背景图像会随着页面其余部分的滚动而移动。
- fixed：当页面的其余部分滚动时，背景图像不会移动。
- inherit：规定应该从父元素继承background-attachment属性的设置。

如何在一个声明中设置所有背景属性：

```
body {
    background: #ff0000 url("../Images/cookies.png") no-repeat fixed center;
}
```

3.4.3 边框属性

border简写属性在一个声明设置所有的边框属性。可以按顺序设置如下属性：

border-width属性

规定边框的宽度。

border-width简写属性为元素的所有边框设置宽度，或者单独地为各边边框设置宽度。只有当边框样式不是none时才起作用。如果边框样式是none，边框宽度实际上会重置为0。不允许指定负长度值。

可能的值：

- thin：定义细的边框。
- medium：默认。定义中等的边框。
- thick：定义粗的边框。
- *length*：允许您自定义边框的宽度。
- inherit：规定应该从父元素继承边框宽度。

一个简单的例子：

```
<div class="border-width-demo"> </div>
```

CSS代码：

```
div.border-width-demo {
    width: 120px;
    height: 80px;
    border-color: #f75555;
    border-style: solid;
    border-width: thin medium thick 10px;
}
```

显示效果：



border-width属性的值出现顺序是顺时针方向，上、右、下、左。如果写三个值时，表示的是上边框宽度、左右边框宽度、下边框宽度；二个值时，表示的是上下变框宽度、左右边框宽度；一个值时，表示的是上下

左右边框的宽度。

border-style属性

用于设置元素所有边框的样式，或者单独地为各边设置边框样式。只有当这个值不是none时边框才可能出现。

案例：



CSS代码：

```
div.border-style-demo {
    width: 120px;
    height: 80px;
    border-color: #f75555;
    border-style: dotted solid double dashed;
    border-width: thin medium thick 10px;
}
```

上边框是点状（dotted），右边框是实线（solid），下边框是双线（double），左边框是虚线（dashed）。如果写三个值时，表示的是上边框样式、左右边框样式、下边框样式；二个值时，表示的是上下边框样式、左右边框样式；一个值时，表示的是上下左右边框的样式。写法：

- border-top-width：设置上边框的宽度。
- border-right-width：设置右边框的宽度。
- border-bottom-width：设置下边框的宽度。
- border-left-width：设置左边框的宽度。

border-color属性

设置四条边框的颜色。此属性可设置1到4种颜色。

`border-color`属性是一个简写属性，可设置一个元素的所有边框中可见部分的颜色，或者为4个边分别设置不同的颜色。

- border-top-style：设置上边框的样式。
- border-right-style：设置右边框的样式。
- border-bottom-style：设置下边框的样式。
- border-left-style：设置左边框的样式。

案例：



CSS代码：

```
div.border-color-demo {
    width: 120px;
    height: 80px;
    border-color: red blue yellow green;
    border-style: dotted solid double dashed;
    border-width: thin medium thick 10px;
}
```

上边框是红色（red），右边框是蓝色（blue）、下边框是黄色（yellow）、左边框是绿色（green）。如果写三个值时，表示的是上边框颜色、左右边框颜色、下边框颜色；二个值时，表示的是上下边框颜色、左右边框颜色；一个值时，表示的是上下左右边框的颜色。写法：

- border-top-color：设置上边框的颜色。
- border-right-color：设置右边框的颜色。
- border-bottom-color：设置下边框的颜色。
- border-left-color：设置左边框的颜色。

可能的值：

- color_name：规定颜色值为颜色名称的边框颜色（比如red）。
- hex_number：规定颜色值为十六进制值的边框颜色（比如#ff0000）。
- rgb_number：规定颜色值为rgb代码的边框颜色（比如rgb(255,0,0)）。
- transparent：默认值。边框颜色为透明。
- inherit：规定应该从父元素继承边框颜色。

这个3.4.2中设置背景颜色一样。

其他书写的格式：

- border-top：在一个声明中设置所有的上边框属性。
- border-right：在一个声明中设置所有的右边框属性。
- border-bottom：在一个声明中设置所有的下边框属性。
- border-left：在一个声明中设置所有的左边框属性。

outline（轮廓）属性

绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。

> 轮廓线不会占据空间，也不一定是矩形。

`outline`简写属性在一个声明中设置所有的轮廓属性。

可以按顺序设置如下属性：

- outline-color：规定边框的颜色。
- outline-style：规定边框的样式。
- outline-width：规定边框的宽度。

值的书写和上述类似，这里直接贴出CSS代码：

```css
div.border-outline-demo {
    padding: 5px;
    margin: 1.25em auto;
    width: 200px;
    font-size: .85em;
    text-align: center;
    border: #f75555 solid 1px;
    outline: #f75555 dotted 2px;
    outline-offset: 5px;
    font-family: "黑体";
}
```

显示效果：

是无意穿堂风，偏偏引山洪。

`outline-offset`属性对轮廓进行偏移，并在边框边缘进行绘制，即outline与border之间的距离。

> 多看阅读器、Lithium中只能显示border属性绘制的边框，outline则无法显示。Gitden显示正常。

border-radius属性

为元素添加圆角边框。`border-radius`属性是一个简写属性，用于设置四个`border-*-radius`属性。语法：

```css
border-radius: 1-4 length | % / 1-4 length | %;
```

> 按此顺序设置每个radiu的四个值。如果省略bottom-left，则与top-right相同。如果省略bottom-right，则与top-left相同。如果省略top-right，则与top-left相同。

可能的值：

- *length*：定义圆角的形状。
- %：以百分比定义圆角的形状。

HTML代码为一个div元素内添加空格，并分别命名样式。4个角分别设置不同的半径，案例：



CSS代码：

```
div.border-radius-demo {
    width: 150px;
    height: 150px;
    background-color: #f75555;
    border-top-left-radius: 20px;
    border-top-right-radius: 30px;
    border-bottom-right-radius: 40px;
    border-bottom-left-radius: 50px;
}
```

左上角圆角半径20像素、右上角圆角半径30像素、右下角圆角半径40像素、左下角圆角半径50像素。也可以简写成如下格式：

```
border-radius: 20px 30px 40px 50px;
```

用百分比画一个圆：



CSS代码：

```
div.border-radius-round {
    width: 150px;
    height: 150px;
    background-color: #f75555;
    border-radius: 50%;
}
```

画一个半圆（以此拓展有下部半圆、左部半圆、右部半圆）：



CSS代码：

```css
div.border-radius-semi-circle {
    width: 150px;
    height: 75px;
    background-color: #f75555;
    border-radius: 75px 75px 0 0;
}
```

还有一些其他效果，可以自己摸索，发挥想象力。

border-image属性

将图片规定为元素的边框。

`border-image`属性是一个简写属性，用于设置以下属性：

- border-image-source：用在边框的图片的路径。
- border-image-slice：图片边框向内偏移。
- border-image-width：图片边框的宽度。
- border-image-outset：边框图像区域超出边框的量。
- border-image-repeat：图像边框是否应平铺（repeated）、铺满（rounded）或拉伸（stretched）。

案例：

秦时明月汉时关，万里长征人未还。

但使龙城飞将在，不教胡马度阴山。

CSS代码：

```css
div.border-image-demo {
    margin: 1em auto;
    width: 240px;
    border: 25px solid transparent;
    border-image: url("../Images/border-image.png") 195 195 stretch;
}
div.border-image-demo p {
    font-size: .85em;
    text-align: center;
}
```

transparent是透明的意思。

解释`border-image`：

①没有单位，专指像素。如这里的195，指的就是195像素。这取决于你的边框图像宽度，可以在Ps中进行测量，选择一个值。

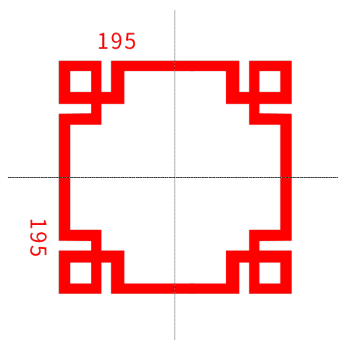


图3-5　border-image调用的图片

②支持百分比值，百分比值大小是相对于边框图片而言，假设边框图片大小为400px*300px，则20%的实际效果就是剪裁了图片的60px　80px　60px 80px的四边大小。

③剪裁特性。从效果的实现上来说，就好像是个剪裁工具，把边框图片四分五裂，再重新安置，变形。其有1～4个参数，其方位规则符合CSS普遍的方位规则，上、右、下、左顺时针，再赋予剪裁的含义。

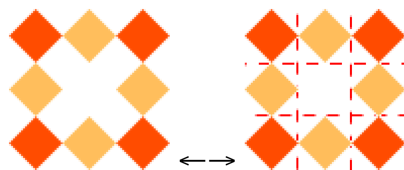


图3-6　border-image裁剪示意

橙红色的四个边角的菱形区域称为「角边框图片」，在border-image中，角边框图片是没有任何展示效果的，不会平铺，不会重复，也不会拉伸，有点类似于视觉中盲点的意思。而对边的四个橙黄色区域属于展示效果的作用区。如果值为repeat，则此区域图片会水平重复；如果是round，则此框框内的图片会水平平铺；如果是stretch，则此矩形域中的图片就会被水平拉伸。左右区域只有垂直方向上的效果，与上下区域效果对应，不多说。



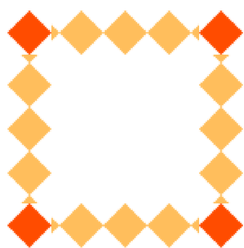




图3-7　border-image裁剪示意

3.4.4 opacity属性

`opacity`属性设置元素的不透明级别。语法：

```
opacity: value | inherit;
```

可能的值：

- *value*：规定不透明度。从0.0 （完全透明）到1.0（完全不透明）。
- inherit：应该从父元素继承opacity属性的值。

案例：

透明度1

透明度0.8

透明度0.6

透明度0.4

透明度0.2

> 在Sigil编辑工具中可以明显看到色块的透明度变化，但是在多看阅读器中显示效果是一样的。

HTML代码：

```
<div class="opacity-1">
  <p>透明度1</p>
</div>
...
<div class="opacity-5">
  <p>透明度0.2</p>
</div>
```

CSS代码：

```
div.opacity-1 {
    width: 20%;
    height: 4em;
    background-color: #ff00ff;
    float: left;
    opacity: 1;
}
```

```
...
div.opacity-5 {
    width: 20%;
    height: 4em;
    background-color: #ff00ff;
    float: left;
    opacity: 0.2;
}
```

从上面的案例可以看到，给div元素设置透明度之后，不光背景颜色变透明了，连div元素里的p元素也透明了。那我们怎么做出一个背景透明而p元素不透明呢？在3.4.2中，我们介绍了一个写法`background-color: rgba(0,0,0,0);`，那我们就可以用这个方法来写一个案例：

背景透明，文字不透明



HTML代码：

```
<div class="opacity-rgba">
  <p>背景透明，文字不透明</p>
  <p><img alt="小可怜儿" src="../Images/kelian.jpg"/></p>
</div>
```

CSS代码：

```
div.opacity-rgba {
    width: 10em;
    background-color: rgba(0,0,0,0.2);
}
div.opacity-rgba img {
    width: 4em;
}
```

背景颜色实际是纯黑色，但因为有设置了透明度，所以显示为灰色。

一个简单的排版案例：

HTML代码：

```html
<div class="opacity-demo">
  <div class="opacity-demo-p">
    <p>【出塞】</p>
    <p>秦时明月汉时关，万里长征人未还。</p>
    <p>但使龙城飞将在，不教胡马度阴山。</p>
  </div>
</div>
```

CSS代码：

```css
div.opacity-demo {
    margin: 1em auto;
    padding: 4em 0;
    width: 300px;
    background-image: url("../Images/poetry-bg.jpg");
    background-repeat: no-repeat;
    background-size: 100% 100%;
}
div.opacity-demo-p {
    padding: .5em .25em;
    width: 80%;
    margin: 0 auto;
    background-color: rgba(0,0,0,0.5);
}
div.opacity-demo-p p {
    font-family: "小标宋";
    font-size: .85em;
    color: #fff;
    text-align: center;
}
```

3.4.5　尺寸属性

height属性

设置元素的高度。

> 这个属性定义元素内容区的高度，在内容区外面可以增加内边距、边框和外边距。行内非替换元素会忽略这个属性。

可能的值：

- auto：默认。浏览器会计算出实际的高度。
- *length*：使用px、em等单位定义高度。
- %：基于包含它的块级对象的百分比高度。
- inherit：规定应该从父元素继承height属性的值。

本例演示如何使用像素值设置元素的高度。



图3-8　img占位图示意

HTML代码：

```
<div class="size-height">
  <img class="big" alt="kelian" src="../Images/kelian.jpg"/>
  <img class="small" alt="kelian" src="../Images/
kelian.jpg"/>
</div>
```

CSS代码：

```
div.size-height {
    margin: 1em auto;
}
div.size-height img.big {
    height: 200px;
}
div.size-height img.small {
    height: 100px;
}
```

width属性

设置元素的宽度。

> 这个属性定义元素内容区的宽度，在内容区外面可以增加内边距、边框和外边距。行内非替换元素会忽略这个属性。

该属性的值设置和`height`属性一样。

CSS盒模型（Box Model）规定了元素框处理元素内容、内边距、边框和外边距的方式。

如果不写任何CSS代码，HTML文档中的元素是按照出现顺序显示的，从上到下，从左到右。这是很自然的处理方式，我们在纸上或者在电脑上写文章也都是按照这种顺序。

具体到一个特定HTML元素的显示，它会被呈现为一个矩形，CSS标准中称之为Box（盒子）。一个HTML元素占据空间的大小由盒模型决定。在盒模型中，一个盒子由内容、内边距、边框和外边距共同构成，其尺寸也是这四部分的尺寸之和。看下面的代码：

```
div.size-box {
    width: 60px;
    height: 60px;
    border: 10px solid #d6005a;
    padding: 30px;
    margin: 40px;
    background-color: #ef92b9;
    page-break-before: always;
}
```

`page-break-before`属性是为了在阅读设备上从新的一页开始，HTML代码：

```
<div class="size-box"> </div>
```

显示效果：



在Chrome的开发者工具中我们可以直观地看到class为box的div元素的盒模型，如下图所示：

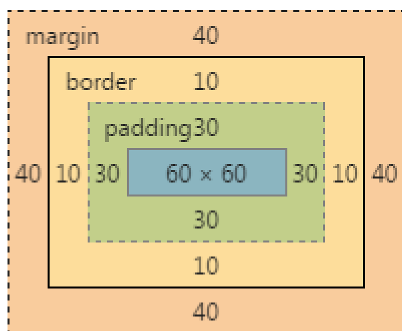


图3-9　盒模型

在上面这个例子中：

1. 内容区域是最内部的浅蓝色方框部分，是由`width: 60px; height: 60px;`定义的，即高为60px，宽为60px
2. 内边距区域由padding属性定义
3. 边框区域由border属性定义
4. 外边距区域由margin属性定义

`margin`和`padding`都有上、右、下、左（顺时针）四个方向的宽度，四个方向的宽度大小可以分别设置。比如对于`padding`有`padding-top`、`padding-right`、`padding-bottom`、`padding-left`，`margin`同理。

注意，例子中`width`属性定义的是内容的宽度，不包含边距、边框。然而在IE中width定义的是内容+内边距+边框的宽度。IE的做法相对更加符

合直观的感觉。width的计算方式是由`box-sizing`属性来定义的，如果设置一个元素为`box-sizing:　border-box;`元素的内边距和边框就会包含在width宽度内。

如果想要页面上所有的元素都以相同的方式定义高度和宽度，可以使用如下代码：

```
* {
    box-sizing: border-box;
}
```

可参考下图：

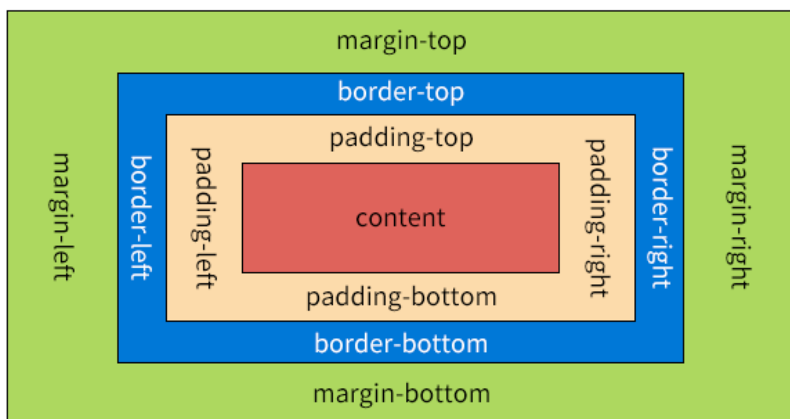


图3-10　IE浏览器下的盒模型

3.4.6 字体属性

`font`简写属性在一个声明中设置所有字体属性。

> 此属性也有第六个值：`line-height`，可设置行间距。

可以按顺序设置如下属性：

font-style属性

`font-style`属性定义字体的样式。

> 该属性设置使用斜体、倾斜或正常字体。斜体字体通常定义为字体系列中的一个单独的字体。

可能的值：

- normal：默认值。浏览器显示一个标准的字体样式。
- italic：浏览器会显示一个斜体的字体样式。
- oblique：浏览器会显示一个倾斜的字体样式。
- inherit：规定应该从父元素继承字体样式。

本例演示如何设置字体风格。

本例演示如何设置字体风格。

浏览器会显示一个斜体的字体样式。

浏览器会显示一个倾斜的字体样式。

CSS代码：

```
p.style-normal {
    font-style: normal;
}
p.style-italic {
    font-style: italic;
}
p.style-oblique {
    font-style: oblique;
}
```

之前在2.2.4中介绍了斜体，这个属于标签本身的样式。按照内容与样式

分离的原则，建议使用`font-style`代替。

下面介绍下italic和oblique的区别。

西文最典型的斜体是italic，它源自一些倾斜的手写体，除了倾斜，还有不少特殊的书法细节。它后来被引入印刷，然后又引入电脑。

而晚些时候出现了一些简单倾斜的西文字体设计，一般称它们为oblique以示区分。Oblique字体的效果往往不及专门设计的斜体，因为它没有被优化，只是把笔画进行倾斜。

浏览器为了体现网页制作者指定的字体italic属性，在没有斜体支持的情况下不得不渲染出简单倾斜的「仿斜体」，类似于oblique，甚至不如预先制作的oblique（因为倾斜角度不一定合适，且预先制作的字体可能会对笔画细节和字体渲染都有一定的优化）。

所以对西文来说，特地设计的斜体是非常有必要的（只要你在乎文本的美观与易读），而且最好是italic。

汉字至今没有广泛认同的与italic搭配的字体样式，汉字字体设计也不会去包含斜体。于是浏览器只好和西文一样去渲染汉字的「仿斜体」，但这个效果往往比西文「仿斜体」还糟糕，主要是因为汉字结构比较复杂，且没有倾斜书写的传统。

所以对汉字来说，斜体一般都是没有必要的，不论是在浏览器里还是别处。

font-weight属性

`font-weight`属性设置文本的粗细。

> 该属性用于设置显示元素的文本中所用的字体加粗。

- normal：默认值。定义标准的字符。
- bold：定义粗体字符。
- bolder：定义更粗的字符。
- lighter：定义更细的字符。
- 100|200|300|400|500|600|700|800|900：定义由粗到细的字符。400等同 normal，而700等同于bold。
- inherit：规定应该从父元素继承字体的粗细。

本例演示如何设置字体的粗细。

本例演示如何设置字体粗细。

浏览器会显示一个加粗的字体样式。

浏览器会显示一个最粗的字体样式。

HTML代码：

```
<div class="test-result">
  <p class="weight-normal">本例演示如何设置字体粗细。</p>
  <p class="weight-bold">浏览器会显示一个加粗的字体样式。</p>
  <p class="weight-900">浏览器会显示一个最粗的字体样式。</p>
</div>
```

CSS代码：

```
p.weight-normal {
    font-weight: normal;
}
p.weight-bold {
    font-weight: bold;
}
p.weight-900 {
    font-weight: 900;
}
```

关于粗体，比较简单：

浏览器生成的「仿粗体」无法像特地设计的粗体一样，在增大笔画粗细的同时保证字体的美观与易读。

它会使字体的风格改变，且破坏了设计师精心调整的细节，降低易读性（甚至笔画可能粘连）。而且，为防止文字加粗后糊成一片，浏览器是不敢把字体加粗太多的，而设计出的粗体能提供更夸张、多种粗细且美观的粗笔画效果。

所以文中如果存在有加粗的内容，可以更换成另一种字体或使用单独设计的不同字重字体。例如思源黑体有7种字重：ExtraLight（特细）、Light、Normal（正常）、Regular、Medium、Bold（粗体）、Heavy（最粗）。



图3-11 思源黑体的7个字重

之前在2.2.3中也有加粗的标签，推荐使用`font-weight`属性代替。

font-size属性

`font-size`属性可设置字体的尺寸。

> 该属性设置元素的字体大小。注意，实际上它设置的是字体中字符框的高度；实际的字符字型可能比这些框高或矮（通常会矮）。

- *length*：把font-size设置为一个固定的值。
- %：把font-size设置为基于父元素的一个百分比值。
- inherit：规定应该从父元素继承字体尺寸。

还有一些其他值，请参考W3School。

本例演示如何设置字体尺寸。

一个固定的值，20px；

一个相对值，1.4em；

一个百分比值，120%。

HTML代码：

```html
<p class="size-length">一个固定的值，20px；</p>
<p class="size-length-em">一个相对值，1.4em；</p>
<p class="size-percent">一个百分比值，120%。</p>
```

CSS代码：

```css
p.size-length {
    font-size: 20px;
}
p.size-length-em {
    font-size: 1.4em;
}
p.size-percent {
    font-size: 120%;
}
```

一般浏览器如果不在body元素里指定具体的字体大小，它的默认大小是16px。那么第一个的意思就很明确；第二个的意思，em是一个相对单位，1.4em指的是16px×1.4，即22.4px；第三个百分比值，就是16px×1.2，即19.2px。你可以用Chrome浏览器打开当前HTML文件，然后使用开发者工具查看具体字体尺寸，下面截取部分：

▶ display	block
▶ font-family	AdobeGa…
▶ font-size	22.4px

图3-12　计算字号后的p标签

上面的截图显示的是1.4em的值实际大小。

如果我们在body里指定`font-size`为12px，根据继承性，则当你设置以上值时，第一个仍是20px，第二个就是12px×1.4＝16.8px，第三个即12px×1.2＝14.4px。

在排版时推荐使用相对单位，而不是绝对值，这样我们只需要调整em参考的`font-size`，就可以改变全文的字体大小。

关于em：不论在Word里、Photoshop里还是web里，缩放字体时我们做的其实就是把em和某个绝对长度对应起来。当em与某个绝对长度有了对应，原本抽象的字体缩放到多大就确定了：如果我们设定字号为12px，那么字体的1em就等于12px。这相当于我们在缩放字体设计的坐标系。

西文字体和em的大小关系其实没有标准。有的字体会相对大一些（比如

Verdana），有的字体会相对小一些（比如 Times New Roman），有的宽，有的窄。了解它们大致的大小就够了。

中文字体和em的关系比较明确。中文字体每个汉字占用的横向空间都是1em。比如，对于12px的汉字文本，10个字排在一起将占用120px的横向空间。但由于汉字本身的面积其实不一样（胖瘦不一样），有的偏大（比如微软雅黑），有的偏小（比如华文仿宋），所以尽管任何字体的10个汉字都会占用120px的横向空间，这10个字的疏密情况不一样。如下：

秦时明月汉时关，万里长征人未还。
但使龙城飞将在，不教胡马度阴山。

— 微软雅黑 —

秦时明月汉时关，万里长征人未还。
但使龙城飞将在，不教胡马度阴山。

— 华文仿宋 —

图3-13　字的疏密

关于CSS的单位见3.5节。

line-height属性

`line-height`属性设置行间的距离（行高）。

- normal：默认。设置合理的行间距。
- number：设置数字，此数字会与当前的字体尺寸相乘来设置行间距。
- *length*：设置固定的行间距。
- %：基于当前字体尺寸的百分比行间距。
- inherit：规定应该从父元素继承line-height属性的值。

本例演示如何使用百分比值来设置段落中的行间距。

世上的武功确实没有高
低之分，只有习武的人
才有强弱之比，通过竞
技，我们可以发现一个
真正的自己，因为我们

真正的对手，可能就是
我们自己。

HTML代码：

```
<div class="line-height-percent">
    <p>世上的武功确实没有高低之分，只有习武的人才有强弱之比，通过竞技，我们可以发现一个真正的自己，因为我们真正的对手，可能就是我们自己。</p>
</div>
```

CSS代码：

```
div.line-height-percent {
    width: 10em;
}
div.line-height-percent p {
    line-height: 90%;
}
```

本例演示如何使用一个相对数值来设置段落中的行间距。

世上的武功确实没有高
低之分，只有习武的人
才有强弱之比，通过竞
技，我们可以发现一个
真正的自己，因为我们
真正的对手，可能就是
我们自己。

HTML代码和上面的例子类似，此处就不在贴出。

CSS代码：

```
div.line-height-em {
    width: 10em;
}
div.line-height-em p {
    line-height: 1.5em;
}
```

对于`line-height`，应当明白它是「行高」而非「行间距」，它是相邻两行文本相同部位的纵向距离（比如，上一行中字母x的左下角到下一行

中字母x的左下角），并非两行文本之间视觉空白的高度。应当了解不同行高时的大致效果（比如西文文本`line-height: 1.2em;`的效果，中文文本`line-height: 1.5em;`的效果……）。

西文的基本行高通常是字号的1.2 上下，而中文因为字符密实且高度一致，没有西文的上伸部（ascender）和下延部（descender）来创造行间空隙，所以一般行高需要更大，可达到1.5至1.8倍甚至更大。

总的来说，字号越小，字距应当相对越大，行高也应该相对越大。反之亦然。具体的数值依不同的媒介、阅读环境、内容、阅读人群和字体等因素而不同。

font-family属性

`font-family`规定元素的字体系列。

`font-family`可以把多个字体名称作为一个「回退」系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。也就是说，`font-family`属性的值是用于某个元素的字体族名称或类族名称的一个优先表。浏览器会使用它可识别的第一个值。

有两种类型的字体系列名称：

1. 指定的系列名称：具体字体的名称，比如宋体、黑体、楷体……
2. 通常字体系列名称：比如serif、sans-serif、cursive、fantasy、monospace……

> 使用逗号分割每个值，并始终提供一个类族名称作为最后的选择。
> 使用某种特定的字体系列完全取决于用户机器上该字体系列是否可用；这个属性没有指示任何字体下载的链接。因此，强烈推荐使用一个通用字体系列名作为后路。

- family-name：用于某个元素的字体族名称或/及类族名称的一个优先表。
- inherit：规定应该从父元素继承字体系列。

比如当前ePub电子书的字体样式表（部分）：

```
@font-face {
    font-family: "小标宋";
    src: local("h3"),local("h3 XiaoBiaoSong"),local("Heading3"),local("Title3"),local("标题3"),
    local("方正小标宋_GBK"),local("方正小标宋简体"),local("方正小标宋繁体"),
    local("Songti"),local("Songti SC"),local("Songti TC"),
    local("STSongti"),local("STSong"),local("Song S"),local("Song T"),
    local("STBShusong"),local("TBMincho"),local("HYMyeongJo"),
    local("DK-XIAOBIAOSONG"),
```

```
    url("../Fonts/XiaoBiaoSong.ttf");
}
```

浏览器会先查找名为h3的字体，如果当前设备中没有，则找下一个h3 XiaoBiaoSong，……，直到找到可用的字体。

利用这个规则，我们可以给中文中的西文进行单独的字体指定，如果你不想使用中文字体里的西文字符。比如当前电子书的正文，中文使用宋体，西文使用AdobeGaramondLT-Regular字体，那么CSS代码：

```
p {
    margin: .5em 0;
    line-height: 1.35em;
    font-family: "AdobeGaramondLT-Regular","宋体";
}
```

如果AdobeGaramondLT-Regular字体当前设备中没有安装，你可以把该字体添加到ePub电子书中的Fonts文件夹中，然后使用`@font-face`语法引入外部字体。具体步骤参考下面的文章。

下面的一个小案例，实现了文字是黑体，而弯双引号是宋体：

```
<p>我是「黑体」，但是我的弯双引号是宋体。</p>
```

CSS代码：

```
@font-face {
    font-family: "黑体";
    src: local("Microsoft Yahei"),
        url("../Fonts/SimHei.ttf");
}
@font-face {
    font-family: "宋体";
    src: local("SimSun"),
        url("../Fonts/SimSun.ttf");
    unicode-range: U+201c, U+201d;
}
p {
    font-family: "宋体", "黑体";
}
```

内嵌过字体之后，在Lithium阅读器中即可实现如上要求。案例下载。

内嵌一个英文花式字体，首先在HTML文件中书写如下：

```
<div class="heather">
```

```
    <p>William Shakespear Sonnets</p>
    <p>Sonnet 18</p>
    <p>Shall I compare thee to a summer's day?</p>
    <p>Thou art more lovely and more temperate:</p>
    <p>Rough winds do shake the darling buds of May,</p>
    <p>And summer's lease hath all too short a date;</p>
    <p>Sometime too hot the eye of heaven shines,</p>
    <p>And often is his gold complexion dimm'd,</p>
    <p>And every fair from fair sometime declines,</p>
    <p>By chance or nature's changing course untrimm'd:</p>
    <p>But thy eternal summer shall not fade</p>
    <p>Nor lose possession of that fair thou ow'st,</p>
    <p>Nor shall Death brag thou wand'rest in his shade,</p>
    <p>When in eternal lines to time thou grow'st.</p>
    <p>So long as men can breathe or eyes can see,</p>
    <p>So long lives this,and this gives life to thee.</p>
</div>
```

然后根据之前学习的属性定义基本的样式，比如字体系列、大小、行间距等。CSS代码：

```
div.heather p {
    margin: .35em 0;
    font-family: "heather-script-two";
    font-size: 1.35em;
    line-height: 1.2em;
}
```

然后使用`@font-face`语法引入外部字体：

```
@font-face {
    font-family: "heather-script-two";
    src: url("../Fonts/heather-script-two.ttf");
}
```

最后显示效果如下：

William Shakespear Sonnets

Sonnet 18

Shall I compare thee to a summer's day?

Thou art more lovely and more temperate:

Rough winds do shake the darling buds of May,

And summer's lease hath all too short a date;

Sometime too hot the eye of heaven shines,

And often is his gold complexion dimm'd;

And every fair from fair sometime declines,

By chance or nature's changing course untrimm'd;

But thy eternal summer shall not fade

Nor lose possession of that fair thou ow'st;

Nor shall Death brag thou wand'rest in his shade,

When in eternal lines to time thou grow'st:

So long as men can breathe or eyes can see,

So long lives this, and this gives life to thee.

在`@font-face`规则中，您必须首先定义字体的名称，然后指向该字体文件的路径。

- font-family（name）：必需。规定字体的名称。这里是`"heather-script-two"`。
- src（URL）：必需。定义字体文件的路径。这里是`"../Fonts/heather-script-two.ttf"`。

关于`@font-face`的使用，还有其他声明，请参考W3School。

3.4.7 列表属性

`list-style`属性在一个声明中设置所有的列表属性。

> 该属性是一个简写属性，涵盖了所有其他列表样式属性。由于它应用到所有display为list-item的元素，所以在普通的HTML和XHTML中只能用于li元素，不过实际上它可以应用到任何元素，并由list-item元素继承。

可以按顺序设置如下属性：

list-style-type属性

设置列表项标记的类型。

可能的值：

- disc：默认。标记是实心圆。
- circle：标记是空心圆。
- square：标记是实心方块。
- decimal：标记是数字。
- decimal-leading-zero：0开头的数字标记。（01, 02, 03, 等。）
- lower-roman：小写罗马数字（i, ii, iii, iv, v, 等。）
- upper-roman：大写罗马数字（I, II, III, IV, V, 等。）
- lower-alpha：小写英文字母（a, b, c, d, e, 等。）
- upper-alpha：大写英文字母（A, B, C, D, E, 等。）
- cjk-ideographic：简单的表意数字。

见2.1.5节。

下面演示几个上文中没有展示的项目标记符：

11. Decimal-leading-zero 类型
12. 茶
13. 可口可乐

N. Lower-roman 类型
O. 茶
P. 可口可乐

q. Upper-roman 类型
r. 茶
s. 可口可乐

- Cjk-ideographic 类型

- 茶
- 可口可乐

HTML代码：

```
<ul style="list-style-type: decimal-leading-zero;">
  <li>Decimal-leading-zero 类型</li>
  <li>茶</li>
  <li>可口可乐</li>
</ul>
<ul style="list-style-type: upper-alpha;">
  <li>Lower-roman 类型</li>
  <li>茶</li>
  <li>可口可乐</li>
</ul>
<ul style="list-style-type: lower-alpha;">
  <li>Upper-roman 类型</li>
  <li>茶</li>
  <li>可口可乐</li>
</ul>
<ul style="list-style-type: cjk-ideographic;">
  <li>Cjk-ideographic 类型</li>
  <li>茶</li>
  <li>可口可乐</li>
</ul>
```

> 多看、掌阅均不支持cjk表意列表。

list-style-position属性

设置在何处放置列表项标记。

该属性用于声明列表标志相对于列表项内容的位置。外部（outside）标志会放在离列表项边框边界一定距离处，不过这距离在CSS中未定义。内部（inside）标志处理为好像它们是插入在列表项内容最前面的行内元素一样。

可能的值：

- inside：列表项目标记放置在文本以内，且环绕文本根据标记对齐。
- outside：默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。
- inherit：规定应该从父元素继承list-style-position属性的值。

本例演示在何处放置列表标记。

该列表的list-style-position的值是inside：

- Earl Grey Tea——一种黑颜色的茶
- Jasmine Tea——一种神奇的「全功能」茶
- Honeybush Tea——一种令人愉快的果味茶

该列表的list-style-position的值是outside：

- Earl Grey Tea——一种黑颜色的茶
- Jasmine Tea——一种神奇的「全功能」茶
- Honeybush Tea——一种令人愉快的果味茶

HTML代码：

```
<p>该列表的list-style-position的值是"inside"：</p>
<ul style="list-style-position: inside;">
  <li>Earl Grey Tea——一种黑颜色的茶</li>
  <li>Jasmine Tea——一种神奇的「全功能」茶</li>
  <li>Honeybush Tea——一种令人愉快的果味茶</li>
</ul>
<p>该列表的list-style-position的值是"outside"：</p>
<ul style="list-style-position: outside;">
  <li>Earl Grey Tea——一种黑颜色的茶</li>
  <li>Jasmine Tea——一种神奇的「全功能」茶</li>
  <li>Honeybush Tea——一种令人愉快的果味茶</li>
</ul>
```

list-style-image属性

使用图像来替换列表项的标记。

> 这个属性指定作为一个有序或无序列表项标志的图像。图像相对于列表项内容的放置位置通常使用`list-style-position`属性控制。
> 请始终规定一个`list-style-type`属性以防图像不可用。

可能的值：

- URL：图像的路径。
- none：默认。无图形被显示。
- inherit：规定应该从父元素继承list-style-image属性的值。

本例演示如何将图像作为列表项标记。

- 咖啡
- 茶
- 可口可乐

HTML代码：

```
<ul class="list-style-image">
  <li>咖啡</li>
  <li>茶</li>
  <li>可口可乐</li>
</ul>
```

CSS代码：

```
ul.list-style-image {
    list-style-type: upper-roman;
    list-style-position: outside;
    list-style-image: url("../Images/title.png");
}
```

3.4.8 外边距属性

`margin`属性在一个声明中设置所有外边距属性。该属性可以有1到4个值。

> 这个简写属性设置一个元素所有外边距的宽度，或者设置各边上外边距的宽度。块级元素的垂直相邻外边距会合并，而行内元素实际上不占上下外边距。行内元素的左右外边距不会合并。同样地，浮动元素的外边距也不会合并。允许指定负的外边距值，不过使用时要小心。
> 允许使用负值。

可能的值：

- auto：浏览器计算外边距。
- *length*：规定以具体单位计的外边距值，比如像素、厘米等。默认值是0px。
- %：规定基于父元素的宽度的百分比的外边距。
- inherit：规定应该从父元素继承外边距。

所有的边距属性在一个声明中

本例演示如何将所有的边距属性设置于一个声明中。

这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落带有指定的外边距。

HTML代码：

```
<p class="margin-demo">这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落带有指定的外边距。</p>
```

CSS代码：

```
p.margin-demo {
    margin: 1.5em 2em 1em 2.5em!important;
}
```

解释：

1. 上外边距 margin-top: 1.5em;（1.5×16＝24px）
2. 右外边距 margin-right: 2em;（2×16＝32px）
3. 下外边距 margin-bottom: 1em;（1×16＝16px）

4. 左外边距 margin-left: 2.5em;（2.5×16＝40px）

em表示是父级元素定义的`font-size`的倍数，这里的父级`font-size`为默认的16px，则盒模型如下：

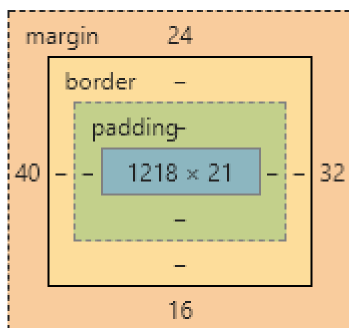


图3-14　margin盒模型

书写4个值表示的是上、右、下左顺时针方向的外边距；书写3个值：

```
margin: 10px 20px 15px;
```

解释：

1. 上外边距 margin-top: 10px;
2. 左右外边距 margin-left: 20px;/margin-right: 20px;
3. 下外边距 margin-bottom: 15px;

书写2个值：

```
margin: 10px 20px;
```

解释：

1. 上下外边距 margin-top: 10px;/margin-bottom: 10px;
2. 左右外边距 margin-left: 20px;/margin-right: 20px;

书写1个值：

```
margin: 10px;
```

所有外边距都是10px。

在电子书排版中，一般用p元素包裹正文部分，那么只需要用`margin`属性的上下边距，就可以设置段落与段落之间的距离。

3.4.9 内边距属性

padding简写属性在一个声明中设置所有内边距属性。

> 这个简写属性设置元素所有内边距的宽度，或者设置各边上内边距的宽度。行内非替换元素上设置的内边距不会影响行高计算；因此，如果一个元素既有内边距又有背景，从视觉上看可能会延伸到其他行，有可能还会与其他内容重叠。元素的背景会延伸穿过内边距。
> 不允许指定负边距值。
> 不允许使用负值。

可能的值：

- auto：浏览器计算内边距。
- *length*：规定以具体单位计的内边距值，比如像素、厘米等。默认值是0px。
- %：规定基于父元素的宽度的百分比的内边距。
- inherit：规定应该从父元素继承内边距。

本例演示使用简写属性将所有的内边距属性设置于一个声明中，可以有1到4个值。

这个单元格的每个边拥有相等的内边距。

HTML代码：

```
<table class="padding-demo">
  <tbody>
    <tr>
      <td>这个单元格的每个边拥有相等的内边距。</td>
    </tr>
  </tbody>
</table>
```

CSS代码：

```
table.padding-demo {
    margin: 1em auto;
    border: 1px solid #333;
} /*margin的左右边距为auto，是为了把表格水平居中显示*/
table.padding-demo td {
    padding: 1.5em!important;
    border: 1px solid #333;
    font-family: "宋体";
```

```
    font-size: .75em;
}
```

在这个案例中，`padding`的上、右、下、左内边距都是1.5em。

当`padding`属性只写3个值时：

```
padding: 1.5em 1em 2em;
```

解释：

1. 上内边距 padding-top: 1.5em;
2. 左右内边距 padding-right: 1em;/padding-left: 1em;
3. 下内边距 padding-bottom: 2em;

当padding属性只写2个值时：

```
padding: 1.5em 2em;
```

解释：

1. 上下内边距 padding-top: 1.5em;/padding-bottom: 1.5em;
2. 左右内边距 padding-right: 2em;/padding-left: 2em;

只写一个值时表示的是上、右、下、左都是一个相同的值。

3.4.10 多列属性

多列属性可以在大屏设备上分割文本，可以配合媒体查询语法使用。

column-count属性

规定元素应该被划分的列数。语法

```
column-count: number | auto;
```

解释：

- number：元素内容将被划分的最佳列数。
- auto：由其他属性决定列数，比如 "column-width"。

案例：

人民网北京8月4日电 （赵超）中国互联网络信息中心（CNNIC）今天在京发布第40次《中国互联网络发展状况统计报告》（以下简称为《报告》）。《报告》显示，截至2017年6月，中国网民规模达到7.51亿，占全球网民总数的五分之一。互联网普及率为54.3%，超过全球平均水平4.6个百分点。网民使用手机上网比例提升至96.3%，移动互联网主导地位强化。

中国网民规模达7.51亿，数字技术助推经济社会转型

截至2017年6月，我国网民规模达到7.51亿，半年共计新增网民1992万人，半年增长率为2.7%。互联网普及率为54.3%，较2016年底提升1.1个百分点。以互联网为代表的数字技术正在加速与经济社会各领域深度融合，成为促进我国消费升级、经济社会转型、构建国家竞争新优势的重要推动力。

手机网民占比达96.3%，移动互联网主导地位强化

截至2017年6月，我国手机网民规模达7.24亿，较2016年底增加2830万人。网民中使用手机上网的比例由2016年底的95.1%提升至96.3%，手机上网比例持续提升。上半年，各类手机应用的用户规模不断上升，场景更加丰富。其中，手机外卖应用增长最为迅速，用户规模达到2.74亿，较2016年底增长41.4%；移动支付用户规模达5.02亿，线下场景使用特点突出，4.63亿网民在线下消费时使用手机进行支付。

> 因为多看阅读器不支持多列属性，所以上面的案例在阅读器中与Sigil的预览效果不同。

将一篇报道分成3列。HTML代码：

```
<div class="column-count-demo">
  <p>人民网北京8月4日电……导地位强化。</p>
  <p style="font-family: 黑体;">中国网民规模达7.51亿，数字技术助推经济社会转型</p>
  <p>截至2017年6月，我国……的重要推动力。</p>
  <p style="font-family: 黑体;">手机网民占比达96.3%，移动互联网主导地位强化</p>
  <p>截至2017年6月，我国手……费时使用手机进行支付。</p>
</div>
```

CSS代码：

```
div.column-count-demo {
    column-count: 3;
    -moz-column-count: 3; /* Firefox */
    -webkit-column-count: 3; /* Safari and Chrome */
}
div.column-count-demo p {
    font-family: "宋体";
}
```

Internet Explorer 10和Opera支持`column-count`属性。

Firefox支持替代的`-moz-column-count`属性。

Safari和Chrome支持替代的`-webkit-column-count`属性。

> Internet Explorer 9以及更早版本的浏览器不支持`column-count`属性。

column-gap属性

规定列之间的间隔。

> 如果列之间设置了`column-rule`，它会在间隔中间显示。

语法:

```
column-gap: length | normal;
```

解释：

- *length*：把列间的间隔设置为指定的长度。
- normal：规定列间间隔为一个常规的间隔。W3C建议的值是1em。

比如上面的案例：

> 人民网北京8月4日电 （赵超）中国互联网络信息中心（CNNIC）今天在京发布第40次《中国互联网络发展状况统计报告》（以下简称为《报告》）。《报告》显示，截至2017年6月，中国网民规模达到7.51亿，占全球网民总数的五分之一。互联网普及率为54.3%，超过全球平均水平4.6个百分点。网民使用手机上网比例提升至96.3%，移动互联网主导地位强化。
>
> 中国网民规模达7.51亿，数字技术助推经济社会转型
>
> 截至2017年6月，我国网民规模达到7.51亿，半年共计新增网民1992万人，半年增长率为2.7%。互联网普及率为54.3%，较2016年底提升1.1个百分点。以互联网为代表的数字技术正在加速与经济社会各领域深度融合，成为促进我国消费升级、经济社会转型、构建国家竞争新优势的重要推动力。
>
> 手机网民占比达96.3%，移动互联网主导地位强化
>
> 截至2017年6月，我国手机网民规模达7.24亿，较2016年底增加2830万人。网民中使用手机上网的比例由2016年底的95.1%提升至96.3%，手机上网比例持续提升。上半年，各类手机应用的用户规模不断上升，场景更加丰富。其中，手机外卖应用增长最为迅速，用户规模达到2.74亿，较2016年底增长41.4%；移动支付用户规模达5.02亿，线下场景使用特点突出，4.63亿网民在线下消费时使用手机进行支付。

CSS代码：

```
div.column-gap-demo {
    column-count: 3;
    column-gap: .5em;
    -moz-column-count: 3;
    -moz-column-gap: .5em;
    -webkit-column-count: 3;
    -webkit-column-gap: .5em;
}
```

这个案例中列与列之间的距离是0.5em。

column-rule属性

规定列之间的宽度、样式和颜色规则。

`column-rule`属性是一个简写属性，用于设置所有`column-rule-*`属性。

`column-rule`属性设置列的宽度、样式和颜色规则。语法：

```
column-rule: column-rule-width column-rule-style column-rule-color;
```

column-rule-width属性设置列之间的宽度规则。语法：

```
column-rule-width: thin | medium | thick | length;
```

解释：

- thin：定义纤细规则。
- medium：定义中等规则。
- thick：定义宽厚规则。
- *length*：规定规则的宽度。

`column-rule-style`属性设置列之间的样式规则。语法：

```
column-rule-style: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset;
```

解释：

- none：定义没有规则。
- dotted：定义点状规则。
- dashed：定义虚线规则。
- solid：定义实线规则。
- double：定义双线规则。

还有其他的取值，请参考W3School。

`column-rule-color`属性设置列之间的颜色规则。语法：

```
column-rule-color: color;
```

解释：

- color：规定颜色规则。16进制、RGB、RGBA等。

案例：

人民网北京8月4日电 （赵超）中国互联网络信息中心（CNNIC）今天在京发布第40次《中国互联网络发展状况统计报告》（以下简称为《报告》）。《报告》显示，截至2017年6月，中国网民规模达到7.51亿，占全球网民总数的五分之一。互联网普及率为54.3%，超过全球平均水平4.6个百分点。网民使用手机上网比例提升至

96.3%，移动互联网主导地位强化。

中国网民规模达7.51亿，数字技术助推经济社会转型

截至2017年6月，我国网民规模达到7.51亿，半年共计新增网民1992万人，半年增长率为2.7%。互联网普及率为54.3%，较2016年底提升1.1个百分点。以互联网为代表的数字技术正在加速与经济社会各领域深度融合，成为促进我国消费升级、经济社会转型、构建国家竞争新优势的重要推动力。

手机网民占比达96.3%，移动互联网主导地位强化

截至2017年6月，我国手机网民规模达7.24亿，较2016年底增加2830万人。网民中使用手机上网的比例由2016年底的95.1%提升至96.3%，手机上网比例持续提升。上半年，各类手机应用的用户规模不断上升，场景更加丰富。其中，手机外卖应用增长最为迅速，用户规模达到2.74亿，较2016年底增长41.4%；移动支付用户规模达5.02亿，线下场景使用特点突出，4.63亿网民在线下消费时使用手机进行支付。

CSS代码：

```
div.column-rule-demo {
    column-count: 3;
    column-gap: 1em;
    column-rule-width: 1px;
    column-rule-style: solid;
    column-rule-color: #f75555;
    -moz-column-count: 3;
    -moz-column-gap: 1em;
    -moz-column-rule-width: 1px;
    -moz-column-rule-style: solid;
    -moz-column-rule-color: #f75555;
    -webkit-column-count: 3;
    -webkit-column-gap: 1em;
    -webkit-column-rule-width: 1px;
    -webkit-column-rule-style: solid;
    -webkit-column-rule-color: #f75555;
}
```

当然也可以使用简写属性（别忘记兼容多浏览器）：

```
column-rule: 1px solid #f75555;
```

column-width属性

规定列的宽度。语法：

```
column-width: auto | length;
```

解释：

- auto：由浏览器决定列宽。
- *length*：规定列的宽度。

比如上述例子：

人民网北京8月4日电 （赵超）中国互联网络信息中心（CNNIC）今天在京发布第40次《中国互联网络发展状况统计报告》（以下简称为《报告》）。《报告》显示，截至2017年6月，中国网民规模达到7.51亿，占全球网民总数的五分之一。互联网普及率为54.3%，超过全球平均水平4.6个百分点。网民使用手机上网比例提升至96.3%，移动互联网主导地位强化。

中国网民规模达7.51亿，数字技术助推经济社会转型

截至2017年6月，我国网民规模达到7.51亿，半年共计新增网民1992万人，半年增长率为2.7%。互联网普及率为54.3%，较2016年底提升1.1个百分点。以互联网为代表的数字技术正在加速与经济社会各领域深度融合，成为促进我国消费升级、经济社会转型、构建国家竞争新优势的重要推动力。

手机网民占比达96.3%，移动互联网主导地位强化

截至2017年6月，我国手机网民规模达7.24亿，较2016年底增加2830万人。网民中使用手机上网的比例由2016年底的95.1%提升至96.3%，手机上网比例持续提升。上半年，各类手机应用的用户规模不断上升，场景更加丰富。其中，手机外卖应用增长最为迅速，用户规模达到2.74亿，较2016年底增长41.4%；移动支付用户规模达5.02亿，线下场景使用特点突出，4.63亿网民在线下消费时使用手机进行支付。

CSS代码：

```
div.column-width-demo {
    column-width: 200px;
    -moz-column-width: 200px;
    -webkit-column-width: 200px;
}
```

3.4.11 定位属性

ePub属于自适应布局排版，所以强烈不建议使用绝对定位等方式进行排版。

ePub先是一个HTML文档，该文档内容有长有短，而在手机阅读器中它是按照一屏一屏展示内容。当你改变字号后，全文字号随之改变，阅读器会重新计算布局。在使用绝对定位或相对定位时，它根据父元素或自身位置做参考。这与之前的布局冲突了，会带来一系列的问题。换句话说，就是页面会崩，内容排版不会按照制作者的意图展示内容。所以在排一些需要绝对定位的页面（篇章页等）时，推荐使用边距法调整或者使用图片进行替换。但以下属性可以使用，增强排版（主要用在图片绕排上）。

float属性

规定框是否应该浮动。

`float`属性定义元素在哪个方向浮动。以往这个属性总应用于图像，使文本围绕在图像周围，不过在 CSS 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。

如果浮动非替换元素，则要指定一个明确的宽度；否则，它们会尽可能地窄。

> 假如在一行之上只有极少的空间可供浮动元素，那么这个元素会跳至下一行，这个过程会持续到某一行拥有足够的空间为止。

可能的值：

- left：元素向左浮动。
- right：元素向右浮动。
- none：默认值。元素不浮动，并会显示在其在文本中出现的位置。
- inherit：规定应该从父元素继承 float 属性的值。

如果想清除浮动对下面元素的影响，可以在浮动对象后面添加一个`<div>`，然后设置`clear:both;`清除浮动。`clear`属性规定元素的哪一侧不允许其他浮动元素。

设置了`float`属性的元素具有以下一些特点：

1. 浮动元素的下一个兄弟元素会紧贴到该元素之前的非浮动元素之后；如果下一个兄弟元素如果也设置了同一浮动方向，则会紧随该元素之后显示。

2. 如果该元素的下一个兄弟元素中有内联元素（通常是文字），则会围绕该元素显示，形成类似「文字围绕图片」的效果。
3. 该元素将变为块级元素，相当于给该元素设置了`display: block;`。

下面我们通过几个例子来看看float会带来什么效果。我们先来展示三个平铺的<div>：



HTML代码：

```
<div class="float-box-blue"> </div>
<div class="float-box-green"> </div>
<div class="float-box-red"> </div>
```

CSS代码：

```
div.float-box-blue {
    margin: 0;
    width: 50px;
    height: 50px;
    background-color: blue;
}
div.float-box-green {
    margin: 0;
    width: 50px;
    height: 50px;
    background-color: green;
}
div.float-box-red {
    margin: 0;
    width: 50px;
    height: 50px;
    background-color: red;
}
```

可以看到蓝色、绿色和红色方块从上往下排列，典型的块元素排列方式。现在让我们把蓝色方块设置为`float: right;`效果如下：



后面直接使用内联式进行样式的修改。

如果将蓝色方块设置为`float: left;`则蓝色方块会遮挡绿色方块，效果如下：



如果三个方块都向左浮动，设置`float: left;`效果如下：



注意此时三个方块的父`<div>`元素是没有高度的，因为三个脱离了普通文档流，父`<div>`元素此时并没有包含这三个方块。如果父`<div>`的宽度不够容纳三个方块，会出现什么情况呢？我们将test-result的width改为140px。



如果将蓝色方块的高度改为120px，则蓝色方块会把红色方块挤向最左侧，效果如下：



如果该元素的下一个兄弟元素中有内联元素（通常是文字），则会围绕该元素显示，形成类似「文字围绕图片」的效果。即虽然浮动会让元素脱离文档流，但是文本内容会受到浮动元素的影响。

虽然蓝色方块浮动起来覆盖了部分绿色方块，在绿色方块中的文字并不会被覆盖。

使用小括号指定一个子表达式后，匹配这

个子表达式的文本可以在表达式或其它程序中作进一步的处理。使用小括号指定一个子表达式后，匹配这个子表达式的文本可以在表达式或其它程序中作进一步的处理。

> 在多看阅读器中，蓝色方块显示为空白。

`clear`属性用以清除浮动带来的影响。比如上一个示例中，为了让绿色方块不被蓝色方块遮挡，可以给绿色方块设置`clear: left;`即清除向左浮动带来的影响。你还可以用right或both来清除向右浮动或同时清除向左向右浮动，表示元素的哪些边不挨着浮动元素。

对元素清理实际上为前面的浮动元素留出了垂直空间，这可以解决我们之前的一个问题。三个`float:  left;`的`<div>`因为不会占据空间，因此它们的父`<div>`高度为0。为了让父`<div>`包裹住三个方块，可以增加一个空的`<div>`并设置`clear: both;`。



HTML代码：

```
<div class="test-result">
  <div style="float: left;" class="float-box-blue"> </div>
  <div style="float: left;" class="float-box-green"> </div>
  <div style="float: left;" class="float-box-red"> </div>
  <div style="clear: both;"></div>
</div>
```

从效果中可以看到，三个色块都包含在父`<div>`中了。

使用浮动不光可以制作绕排，还可以做首字大写：

这这是一沟绝望的死水，清风吹不起半点漪沦。不如多扔些破铜烂铁，爽性泼你的剩菜残羹。也许铜的要绿成翡翠，铁罐上锈出几瓣桃花；再让油腻织一层罗绮，霉菌给他蒸出些云霞。让死水酵成

一沟绿酒，漂满了珍珠似的白沫；小珠们笑声变成大珠，又被偷酒的花蚊咬破。那么一沟绝望的死水，也就夸得上几分鲜明。如果青蛙耐不住寂寞，又算死水叫出了歌声。这是一沟绝望的死水，这里断不是美的所在，不如让给丑恶来开垦，看它造出个什么世界。

HTML代码：

```
<div class="test-result">
    <p><span class="first-word">这</span>这是一沟绝望的死水，清风吹不起……世界。</p>
</div>
```

CSS代码：

```
div.test-result span.first-word {
    margin-right: 3px;
    float: left;
    font-size: 2em;
    line-height: .9em;
    font-family: "黑体";
}
```

关于图片绕排可能带来的一些问题，见10.4.6节。

vertical-align属性

设置元素的垂直对齐方式。

> 该属性定义行内元素的基线相对于该元素所在行的基线的垂直对齐。允许指定负长度值和百分比值。这会使元素降低而不是升高。在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。

可能的值：

- baseline：默认。元素放置在父元素的基线上。
- sub：垂直对齐文本的下标。
- super：垂直对齐文本的上标。
- top：把元素的顶端与行中最高元素的顶端对齐。
- text-top：把元素的顶端与父元素字体的顶端对齐。
- middle：把此元素放置在父元素的中部。
- bottom：把元素的顶端与行中最低的元素的顶端对齐。
- text-bottom：把元素的底端与父元素字体的底端对齐。
- *length*
- %：使用line-height属性的百分比值来排列此元素。允许使用负值。

- inherit：规定应该从父元素继承vertical-align属性的值。

下面我们来看一张图来更好理解垂直对齐主要属性值的表现形式：

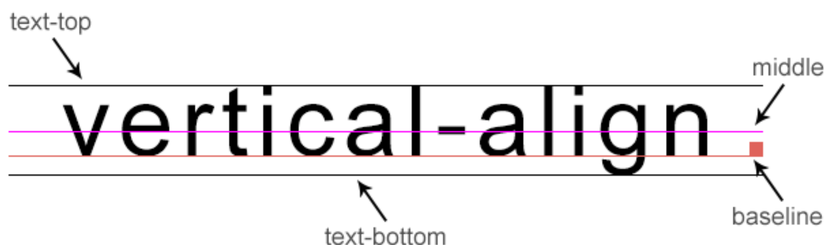


图3-15 vertical-align的属性示意

下面的案例展示了`vertical-align`属性的某些特性：

vertical-align

HTML代码：

```
<p class="vertical-test">vertical-
align<img alt="square" src="../Images/square.png"/></p>
```

CSS代码：

```
p.vertical-test {
    font-size: 2em;
    line-height: 100%;
    font-family: "黑体", Arial;
}
p.vertical-test img {
    vertical-align: bottom;
}
```

你可以更改`vertical-align`的值来查看这些值的实际显示效果。

本例演示如何在文本中垂直排列图象。

这是一幅位于段落中的图像，对齐文本顶部。

这是一幅位于段落中的图像，对齐文本下部。

> 在多看阅读器中两张行内图显示一样，Lithium、Gitden正常。

HTML代码：

```
<p>这是一幅<img class="vertical-text-top" alt="kelian" src="../Images/kelian.jpg"/>位于段落中的图像，对齐文本顶部。</p>
<p>这是一幅<img class="vertical-text-bottom" alt="kelian" src="../Images/kelian.jpg"/>位于段落中的图像，对齐文本下部。</p>
```

CSS代码：

```
img.vertical-text-top {
    width: 2em;
    vertical-align: text-top;
}
img.vertical-text-bottom {
    width: 2em;
    vertical-align: text-bottom;
}
```

当你在表格中使用垂直对齐时记得要设置垂直对齐方式。

3.4.12 打印属性

`page-break-after`属性设置元素后的分页行为，`page-break-before`属性设置元素前的分页行为。

> 请尽可能少地使用分页属性，并且避免在表格、浮动元素、带有边框的块元素中使用分页属性。

应用于：position值为relative或static的非浮动块级元素。

可能的值：

- always：在元素后插入分页符。

还有一些其他值，可以参考W3School。

比如一个章节分为一二三四小节（内容不多），如果用拆分功能把这个章节拆分成4个HTML文件，在文档形式上分开，带来的问题就是文件名需要重命名，表明这是同一个章节的内容。这样做，比较麻烦，现在我们就可以使用`page-break-after:always;`或`page-break-before:always;`来进行显示上的区分（针对阅读系统上的分屏显示，一屏就是一页）。但是在结构上还是属于同一文档。

在阅读器上这个段落将会在新的页面中显示，无论上一页是否有足够的空间来显示。

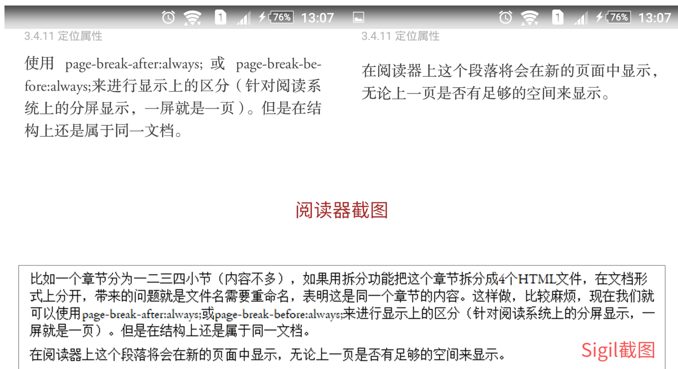


图3-16　page-break属性的应用

从截图中看上一页是有足够的空间来显示这个段落，但是因为我们设置了分页属性，这个段落就从新的一页显示了。而在Sigil里预览是没有分页行为的。上面测试案例的代码：

```
<p style="page-break-before: always;">在阅读器上这个段落将会在新的页面中显示，无论上一页是否有足够的空间来显示。</p>
```

另页起

要使块内容的排版做到在当前页有足够空间显示时，正常显示；当前页不足以显示完整的块内容，另页显示。避免在内容内部进行拆分显示。此时可以使用`page-break-inside: avoid;`声明，设置在元素内部避免进行分页的分页行为。示例代码：

```
<p>后连续耕种的田。……这是村公社可耕的三个部分相等的田，第—年如此。第二年仍耕这三部分田，不过其中菑、新、畬已转为新、畬、菑。同例，第三年又转为畬、菑、新。如图：</p>
  <div class="Figure">
    <p>第一年　菑　新　畬</p>
    <p>　　　　↓　↓　↓</p>
    <p>第二年　新　畬　菑</p>
    <p>　　　　↓　↓　↓</p>
```

```
    <p>第三年　畬　菑　新</p>
  </div>
<p>这就是《尔雅·释地》「田一岁曰菑，二岁曰新田，三岁曰畬」的正解。……这种三田制，在中国古代，是与一年耕百亩休百亩、耕百亩休二百亩的二田制、复田制，同称为爰田。爰、辕、□、换四字，古代音同、义同，故相通用。</p>
```

相关CSS：

```
div.Figure {
    margin: 0.4em 0;
    page-break-inside: avoid;
}
div.Figure p {
    margin: 0;
    text-align: center;
    line-height: 1.25em;
}
```

比如在阅读器中增大字号，内容所占区域也会变大，需要更多的空间。由于无法在当前页完整显示，固定排版内容就会顺延到下一页显示。详细介绍见@云轩阁主的关于电子书中换页的技巧。

3.4.13 表格属性

对表格的布局，可见2.1.4节。

border-collapse属性

设置表格的边框是否被合并为一个单一的边框，还是像在标准的HTML中那样分开显示。

可能的值：

- separate：默认值。边框会被分开。不会忽略border-spacing和empty-cells属性。
- collapse：如果可能，边框会合并为一个单一的边框。会忽略border-spacing和empty-cells属性。
- inherit：规定应该从父元素继承border-collapse属性的值。

本例演示把表格边框显示为一条单独的边框。

五行	方位	五色	五味	五脏
金	西	白	辛	肺
木	东	青	酸	肝
水	北	黑	咸	肾
火	南	赤	苦	心
土	中	黄	甘	脾

HTML代码：

```
<table class="table-collapse-demo">
  <thead>
    <tr>
      <th>五行</th>
      <th>方位</th>
      <th>五色</th>
      <th>五味</th>
      <th>五脏</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>金</td>
      <td>西</td>
      <td>白</td>
      <td>辛</td>
      <td>肺</td>
```

```
    </tr>
    <tr>
      <td>木</td>
      <td>东</td>
      <td>青</td>
      <td>酸</td>
      <td>肝</td>
    </tr>
    <tr>
      <td>水</td>
      <td>北</td>
      <td>黑</td>
      <td>咸</td>
      <td>肾</td>
    </tr>
    <tr>
      <td>火</td>
      <td>南</td>
      <td>赤</td>
      <td>苦</td>
      <td>心</td>
    </tr>
    <tr>
      <td>土</td>
      <td>中</td>
      <td>黄</td>
      <td>甘</td>
      <td>脾</td>
    </tr>
  </tbody>
</table>
```

CSS代码：

```
table.table-collapse-demo {
    margin: 1em auto;
    border: 1px solid #333;
    border-collapse: collapse;
}
table.table-collapse-demo td,th {
    padding: .15em;
    text-align: center;
    vertical-align: middle;
    border: 1px solid #666;
    font-family: "宋体";
}
```

在多看阅读器中，合并的两条边框线并不完全重合而是紧贴在一起。所以当你设置边框为1px，合并后边框变成2px，而不是在Sigil预览中那样是1px。这种问题解决方法如下：

合并边框		
A	B	C
X	Y	Z

CSS代码如下：

```
table.collapse {
    width: 40%;
    border-collapse: collapse;
    margin: .5em auto;
    border-left: 1px solid #000;
    border-top: 1px solid #000;
    border-right: 2px solid #000;
    border-bottom: 2px solid #000;
}
table.collapse th {
    padding: .25em;
    border-left: 1px solid #000;
    border-top: 1px solid #000;
}
table.collapse td {
    text-align: center;
    vertical-align: middle;
    padding: .25em;
    border-left: 1px solid #000;
    border-top: 1px solid #000;
}
```

以上代码在多看中，顶线、底线、墙线会以粗细2px显示，行线以粗线1px显示。

合并边框		
A	B	C
X	Y	Z

图3-17　多看阅读显示的合并边线效果

border-spacing属性

设置相邻单元格的边框间的距离（仅用于「边框分离」模式）。

> 该属性指定分隔边框模型中单元格边界之间的距离。在指定的两个长度值中，第一个是水平间隔，第二个是垂直间隔。除非`border-collapse`被设置为separate，否则将忽略这个属性。尽管这个属性只应用于表，不过它可以由表中的所有元素继承。

可能的值：

- *length length*：规定相邻单元的边框之间的距离。使用px、em等单位。不允许使用负值。如果定义一个length参数，那么定义的是水平和垂直间距。如果定义两个length参数，那么第一个设置水平间距，而第二个设置垂直间距。
- inherit：规定应该从父元素继承border-spacing属性的值。

本例演示如何设置单元格边框之间的距离。

世	界
大	同

HTML代码：

```
<table class="table-spacing-demo">
  <tbody>
    <tr>
      <td>世</td>
      <td>界</td>
    </tr>
    <tr>
      <td>大</td>
      <td>同</td>
    </tr>
```

```
  </tbody>
</table>
```

CSS代码：

```
table.table-spacing-demo {
    margin: 1em auto;
    border: 1px solid #333;
    border-spacing: 2em 1em;
}
table.table-spacing-demo td {
    text-align: center;
    vertical-align: middle;
    border: 1px solid #777;
    font-family: "宋体";
}
```

这个CSS设置了边框与边框之间水平距离2em，垂直距离1em。

3.4.14 文本属性

文本属性一般指的是字符间距、段间距、缩进等属性。

color属性

设置文本的颜色。

这个属性设置了一个元素的前景色（在HTML表现中，就是元素文本的颜色）；光栅图像不受`color`影响。这个颜色还会应用到元素的所有边框，除非被`border-color`或另外某个边框颜色属性覆盖。

要设置一个元素的前景色，最容易的方法是使用`color`属性。

> 请使用合理的背景颜色和文本颜色搭配，这样可以提高文本的可读性。

可能的值：

- color_name：规定颜色值为颜色名称的颜色（比如red）。
- hex_number：规定颜色值为十六进制值的颜色（比如#ff0000）。
- rgb_number：规定颜色值为rgb代码的颜色（比如rgb(255,0,0)）。
- inherit：规定应该从父元素继承颜色。

本例演示如何设置文本的颜色。

使用颜色名设置文本颜色。

使用十六进制值设置文本颜色。

使用rgb代码设置文本颜色。

HTML代码：

```
<p style="color: brown;">使用颜色名设置文本颜色。</p>
<p style="color: #a52a2a;">使用十六进制值设置文本颜色。</p>
<p style="color: rgb(165,42,42);">使用rgb代码设置文本颜色。</p>
```

CSS常用的颜色名、十六进制值、RGB颜色查询对照表、日本传统色。关于颜色的相关知识，在前文中有叙述，在3.4.2一节。

letter-spacing属性

增加或减少字符间的空白（字符间距）。该属性定义了在文本字符框之间插入多少空间。由于字符字型通常比其字符框要窄，指定长度值时，

会调整字母之间通常的间隔。因此，normal就相当于值为0。

> 允许使用负值，这会让字母之间挤得更紧。

可能的值：

- normal：默认。规定字符间没有额外的空间。
- *length*：定义字符间的固定空间（允许使用负值）。
- inherit：规定应该从父元素继承letter-spacing属性的值。

本例演示如何增加或减少字符间距。

Preface

前言

HTML代码：

```
<p class="letter-spacing-demo">Preface</p>
<p class="letter-spacing-demo">前言</p>
```

CSS代码：

```
p.letter-spacing-demo {
    letter-spacing: .5em;
}
```

line-height属性

设置行间的距离（行高）。

> 不允许使用负值。

见3.4.6一节。

text-align属性

规定元素中的文本的水平对齐方式。

该属性通过指定行框与哪个点对齐，从而设置块级元素内文本的水平对齐方式。通过允许用户代理调整行内容中字母和字之间的间隔，可以支持值justify；不同浏览器可能会得到不同的结果。

可能的值：

- left：把文本排列到左边。默认值：由浏览器决定。

- right：把文本排列到右边。
- center：把文本排列到中间。
- justify：实现两端对齐文本效果。
- inherit：规定应该从父元素继承text-align属性的值。

值justify

最后一个水平对齐属性是justify，它会带来自己的一些问题。

值justify可以使文本的两端都对齐。在两端对齐文本中，文本行的左右两端都放在父元素的内边界上。然后，调整单词和字母间的间隔，使各行的长度恰好相等。您也许已经注意到了，两端对齐文本在打印领域很常见。不过在CSS中，还需要多做些考虑。

要由浏览器（而不是CSS）来确定两端对齐文本如何拉伸，以填满父元素左右边界之间的空间。例如，有些浏览器可能只在单词之间增加额外的空间，而另外一些浏览器可能会平均分布字母间的额外空间（不过CSS规范特别指出，如果`letter-spacing`属性指定为一个长度值，「浏览器不能进一步增加或减少字符间的空间」）。还有一些用户代理可能会减少某些行的空间，使文本挤得更紧密。所有这些做法都会影响元素的外观，甚至改变其高度，这取决于用户代理的对齐选择影响了多少文本行。

CSS也没有指定应当如何处理连字符注。大多数两端对齐文本都使用连字符将长单词分开放在两行上，从而缩小单词之间的间隔，改善文本行的外观。不过，由于CSS没有定义连字符行为，浏览器不太可能自动加连字符。因此，在CSS中，两端对齐文本看上去没有打印出来好看，特别是元素可能太窄，以至于每行只能放下几个单词。当然，使用窄设计元素是可以的，不过要当心相应的缺点。

本例演示如何对齐文本。

文本默认。左对齐。

文本右对齐。

文本居中对齐。

文本两端对齐。在排版中也称作齐头尾，即文本区块的左边缘和右边缘都是齐整的。但由于有避头尾禁则和英文分词（hyphenation）的存在，总是需要调节文本间距。一种观点认为字间距应永远保持不变，只通过压缩标点来调节。这样不但不会出现很松的一段字，也可以尽量避免打断「行气」，即出现英文排版术语里所说的river。

在多看阅读器中即使没有设置两端对齐，阅读器也会帮你自动进行对齐，至于对齐的效果完全取决于排版引擎的优化，有的很丑，有的效果特别差，尤其是当一行中有比较长的英文单词时。

避头尾，指点号（顿号、逗号、句号等）、结束引号、结束括号等，不能出现在一行的开头；开始引号、开始括号、开始双书名号等，不能出现在一行的结尾。我记得在上学时写作文，语文老师都有强调过。

text-decoration属性

规定添加到文本的修饰。

> 修饰的颜色由`color`属性设置。
> 这个属性允许对文本设置某种效果，如加下划线。如果后代元素没有自己的装饰，祖先元素上设置的装饰会「延伸」到后代元素中。不要求浏览器支持blink。

可能的值：

- none：默认。定义标准的文本。
- underline：定义文本下的一条线。
- overline：定义文本上的一条线。
- line-through：定义穿过文本下的一条线。
- blink：定义闪烁的文本。
- inherit：规定应该从父元素继承text-decoration属性的值。

本例演示如何向文本添加修饰。

蓝色与下划线是超链接的默认样式，使用none可以去除。本书中超链接的下划线已经去除了。

显示效果与`<u>`标签类似，推荐使用CSS属性代替。

定义文本上的一条线，也可以用border属性定义。

定义穿过文本下的一条线，与`<del>`标签显示类似。

闪烁的文本。

HTML代码：

```
<p>蓝色与下划线是超链接的默认样式，使用none可以去除。本书中超链接的下划线已经去除了。</p>
<p style="text-decoration: underline;">显示效果与<a href="../Text/Chapter2-2.xhtml#html_inline_u">&lt;u&gt;标签</a>类似，推荐使用CSS属性代替。</p>
<p style="text-decoration: overline;">定义文本上的一条线，也可以用<a href="../Text/
```

```
Chapter3-4-3.xhtml#css_attribute_border">border属性</a>定义。</p>
<p style="text-decoration: line-through;">定义穿过文本下的一条线，与<a href="../Text/Chapter2-2.xhtml#html_inline_del">&lt;del&gt;标签</a>显示类似。</p>
<p style="text-decoration: blink;">闪烁的文本。</p>
```

超链接去除下划线CSS：

```
a {
    text-decoration: none;
}
```

text-indent属性

规定文本块中首行文本的缩进。

> 允许使用负值。如果使用负值，那么首行会被缩进到左边。
> 用于定义块级元素中第一个内容行的缩进，指定负值后，会产生一种「悬挂缩进」的效果。

可能的值：

- *length*：定义固定的缩进。默认值：0。
- %：定义基于父元素宽度的百分比的缩进。
- inherit：规定应该从父元素继承text-indent属性的值。

本例演示如何缩进文本首行。

段首缩进两个汉字（的宽度）并不是绝对正确的真理，只是一种习惯。可以缩进一个汉字、两个汉字，也可以不缩进。

手写和纸质印刷中段首缩进两个汉字仍然是约定俗成，很少有人打破；而网页中不用缩进已经可以被绝大多数人都接受，但要配合使用较大的段落间距。

但不论如何都要采取某种措施来作为段落分隔的标志，所以段首缩进和段间距至少要用一种。

HTML代码：

```
<p style="text-indent: 2em;">段首缩进两个汉字（的宽度）并……可以不缩进。</p>
<p style="margin-left: 2em; text-indent: -2em;">手写和纸质印刷中段首缩进两个汉字仍然是约定……段落间距。</p>
<p style="text-indent: 10%;">但不论如何都要采……距至少要用一种。</
```

```
p>
```

text-transform属性

控制文本的大小写。

这个属性用来处理西文的，中文的无用。

> 这个属性会改变元素中的字母大小写，而不论源文档中文本的大小写。如果值为capitalize，则要对某些字母大写，但是并没有明确定义如何确定哪些字母要大写，这取决于浏览器如何识别出各个「词」。
> 不同的用户代理可能会用不同的方法来确定单词从哪里开始，相应地确定哪些字母要大写。例如，文本「w3-school」可以用两种方式显示：「W3-school」和「W3-School」。CSS并没有规定哪一种是正确的，所以这两种都是可以的。

可能的值：

- none：默认。定义带有小写字母和大写字母的标准的文本。
- capitalize：文本中的每个单词以大写字母开头。
- uppercase：定义仅有大写字母。
- lowercase：定义无大写字母，仅有小写字母。
- inherit：规定应该从父元素继承text-transform属性的值。

本例演示如何控制文本中的字母的大小写。

To be or not to be. That is a question.

To be or not to be. That is a question.

To be or not to be. That is a question.

> 多看阅读器并不支持`text-transform`属性，所以上面案例都将显示为第1行的效果。

HTML代码：

```
<p class="text-capitalize">To be or not to be. That is a question.</p>
<p class="text-uppercase">To be or not to be. That is a question.</p>
<p class="text-lowercase">To be or not to be. That is a question.</p>
```

CSS代码：

```
p.text-capitalize {
    text-transform: capitalize;
```

```
} /*每个单词字母开头大写*/
p.text-uppercase {
    text-transform: uppercase;
} /*字母全部大写*/
p.text-lowercase {
    text-transform: lowercase;
} /*字母全部小写*/
```

white-space属性

设置如何处理元素内的空白。

可能的值：

- normal：默认。空白会被浏览器忽略。
- nowrap：文本不会换行，文本会在在同一行上继续，直到遇到`<br>`标签为止。

其他值参考W3School。

word-spacing属性

增加或减少单词间的空白（即字间隔）。

该属性定义元素中字之间插入多少空白符。针对这个属性，「字」 定义为由空白符包围的一个字符串。如果指定为长度值，会调整字之间的通常间隔；所以normal就等同于设置为0。允许指定负长度值，这会让字之间挤得更紧。

> 允许使用负值。
> CSS把「字（word）」定义为任何非空白符字符组成的串，并由某种空白字符包围。这个定义没有实际的语义，它只是假设一个文档包含由一个或多个空白字符包围的字。支持CSS的浏览器不一定能确定一个给定语言中哪些是合法的字，而哪些不是。尽管这个定义没有多大价值，不过它意味着采用象形文字的语言或非罗马书写体往往无法指定字间隔。利用这个属性，可能会创建字间隔太宽的文档，所以，使用`word-spacing`时要小心。

可能的值：

- normal：默认。定义单词间的标准空间。
- *length*：定义单词间的固定空间。
- inherit：规定应该从父元素继承word-spacing属性的值。

案例：

Preface Content

前言 正文

HTML代码：

```
<p class="word-spacing-demo">Preface Content</p>
<p class="word-spacing-demo">前言 正文</p>
```

CSS代码：

```
p.word-spacing-demo {
    word-spacing: 2em;
}
```

text-emphasis属性

该属性可以为文本添加重点标记，位置通常是是位于文本下方。简写属性，用于在一个声明中设置`text-emphasis-style`和`text-emphasis-color`。语法：

```
text-emphasis: text-emphasis-style text-emphasis-color;
```

解释：

- text-emphasis-style：向元素的文本应用重点标记。
- text-emphasis-color：定义重点标记的前景色。
- text-emphasis-positon：设置重点标记的显示位置。

案例：

从小丘西行百二十步，隔篁竹，闻水声，如鸣珮环，心乐之。伐竹取道，下见小潭，水尤清冽。全石以为底，近岸，卷石底以出，为坻，为屿，为嵁，为岩。青树翠蔓，蒙络摇缀，参差披拂。

HTML代码：

```
<p>从<span class="filledDot">小丘</span>西行百二十步，隔<span class="openDot">篁竹</span>，闻水声，如鸣<span class="filledCircle">珮环</span>，心乐之。<span class="openCircle">伐竹取道</span>，下见<span class="filledDouble-circle">小潭</span>，水尤<span class="openDouble-circle">清冽</span>。<span class="filledTriangle">全石</span>以为底，近岸，<span class="openTriangle">卷石</span>底以出，为坻，为屿，为嵁，为岩。<span class="string">青树</span>翠蔓，蒙络摇缀，参差披拂。</p>
```

CSS代码：

```css
/*实心点*/
.filledDot {
    text-emphasis-style: filled dot;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: filled dot;
}
/*空心点*/
.openDot {
    text-emphasis-style: open dot;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: open dot;
}
/*实心圆*/
.filledCircle {
    text-emphasis-style: filled circle;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: filled circle;
}
/*空心圆*/
.openCircle {
    text-emphasis-style: open circle;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: open circle;
}
/*实心同心圆*/
.filledDouble-circle {
    text-emphasis-style: filled double-circle;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: filled double-circle;
}
/*空心同心圆*/
.openDouble-circle {
```

```
    text-emphasis-style: open double-circle;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: open double-circle;
}
/*实心三角*/
.filledTriangle {
    text-emphasis-style: filled triangle;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: filled triangle;
}
/*空心三角*/
.openTriangle {
    text-emphasis-style: open triangle;
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: open triangle;
}
/*字符*/
.string {
    text-emphasis-style: 'O';
    text-emphasis-color: #000;
    text-emphasis-position: under;
    -webkit-text-emphasis-color: #000;
    -webkit-text-emphasis-position: under;
    -webkit-text-emphasis: 'O';
}
```

当不设置`text-emphasis-position: under;`时，标记符会出现在字符上方。请务必带上`-webkit-`前缀。

text-justify属性

规定当`text-align`被设置为justify时的齐行方法。该属性规定如何对齐行文本进行对齐和分隔。语法：

```
text-justify: auto|inter-word|inter-ideograph|inter-cluster|distribute|kashida|trim;
```

解释：

- auto：浏览器决定齐行算法。
- none：禁用齐行。
- inter-word：增加/减少单词间的间隔。
- inter-ideograph：用表意文本来排齐内容。
- inter-cluster：只对不包含内部单词间隔的内容（比如亚洲语系）进行排齐。
- distribute：类似报纸版面，除了在东亚语系中最后一行是不齐行的。
- kashida：通过拉伸字符来排齐内容。

显示效果：

世上的武功确实没有高低之分，只有习武的人才有强弱之比，通过竞技，我们可以发现一个真正的自己，因为我们真正的对手，可能就是我们自己。The world of martial arts really no high and low points, only martial arts talent than the strength of the ratio, through the competition, we can find a real self, because we really opponents, may be our own.

世上的武功确实没有高低之分，只有习武的人才有强弱之比，通过竞技，我们可以发现一个真正的自己，因为我们真正的对手，可能就是我们自己。The world of martial arts really no high and low points, only martial arts talent than the strength of the ratio, through the competition, we can find a real self, because we really opponents, may be our own.

世上的武功确实没有高低之分，只有习武的人才有强弱之比，通过竞技，我们可以发现一个真正的自己，因为我们真正的对手，可能就是我们自己。The world of martial arts really no high and low points, only martial arts talent than the strength of the ratio, through the competition, we can find a real self, because we really opponents, may be our own.

世上的武功确实没有高低之分，只有习武的人才有强弱之比，通过竞技，我们可以发现一个真正的自己，因为我们真正的对手，可能就是我们自己。The world of martial arts really no high and low points, only martial arts talent than the strength of the ratio, through the competition, we can find a real self, because we really opponents, may be our own.

世上的武功确实没有高低之分，只有习武的人才有强弱之比，通过竞技，我们可以发现一个真正的自己，因为我们真正的对手，可能就是我们自己。The world of martial arts really no high and low points, only martial arts talent than the strength of the ratio, through the competition, we can find a real self, because we really opponents, may be our own.

HTML代码：

```
<p class="inter-word-demo">世上的武功确……may be our own.</p>
<p class="inter-ideograph-demo">世上的武功确……may be our own.</p>
<p class="inter-cluster-demo">世上的武功确……may be our own.</p>
<p class="distribute-demo">世上的武功确……may be our own.</p>
<p class="kashida-demo">世上的武功确……may be our own.</p>
```

CSS代码：

```
p.inter-word-demo {
    width: 90%;
    text-align: justify;
    text-justify: inter-word;
} /*增加或减少单词间的间隔*/
p.inter-ideograph-demo {
    width: 80%;
    text-align: justify;
    text-justify: inter-ideograph;
} /*用表意文本来排齐内容*/
p.inter-cluster-demo {
    width: 70%;
    text-align: justify;
    text-justify: inter-cluster;
} /*只对CJK语言内容排齐*/
p.distribute-demo {
    width: 80%;
    text-align: justify;
    text-justify: distribute;
} /*两端对齐，最后一行左对齐*/
p.kashida-demo {
    width: 90%;
    text-align: justify;
    text-justify: kashida;
} /*通过拉伸字符来排齐内容*/
```

表意文字是一种用象征性书写符号记录词或词素的文字体系，不直接或不单纯表示语音。表意文字是一种图形符号只代表语素，而不代表音节的文字系统。表意文字是文字萌芽时期的产物。汉字属于表意文字。——百度百科

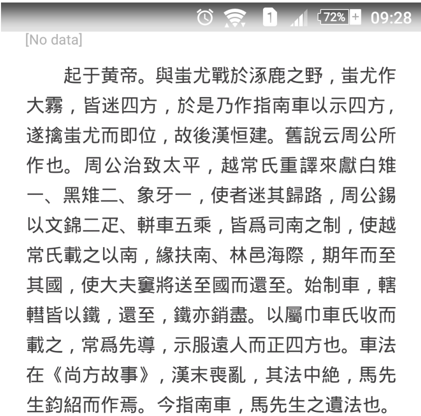
在多看阅读器中没有`text-justify:   inter-ideograph;`这个声明，也没有关系，阅读系统已经默认做好两端对齐了。但是对Lithium来说，添加这个声明是非常有必要的。下面用一段代码展示下：

```
<p class="inter-ideograph">起于黄帝。與蚩尤戰於涿鹿之野，蚩尤作大霧，皆迷四方，於是乃作指南車以示四方，遂擒蚩尤而即位，故後漢恒建。舊說云周公所作也。周公治致太平，越常氏重譯來獻白雉一、黑雉二、象牙一，使者迷其歸路，周公錫以文錦二疋、軿車五乘，皆爲司南之制，使越常氏載之以南，緣扶南、林邑海際，期年而至其國，使大夫宴將送至國而還至。始制車，轄轊皆以鐵，還至，鐵亦銷盡。以屬巾車氏收而載之，常爲先導，示服遠人而正四方也。車法在《尚方故事》，漢末喪亂，其法中絶，馬先生鈞紹而作焉。今指南車，馬先生之遺法也。</p>
```

CSS代码：

```
p.inter-ideograph {
    text-indent: 2em;
    line-height: 1.45em;
}
```

下面两张图分别在多看阅读（左图）和Lithium（右图）中显示如下：



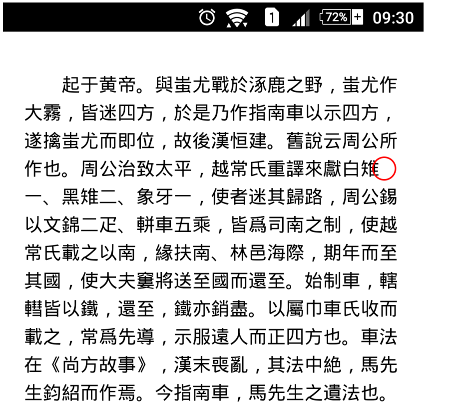


图3-19 文本设置justify时Lithium的显示效果

可以看到在Lithium红圈位置存在空白，这就是没有两端对齐。然后CSS修改成如下：

```
p.inter-ideograph {
    text-indent: 2em;
    line-height: 1.45em;
    text-align: justify;
    text-justify: inter-ideograph;
}
```

这在Lithium中显示如下：



图3-20　Lithium中实现齐头尾、避头尾

添加`text-align: justify;`和`text-justify: inter-ideograph;`之后才实现了两端对齐的目的。

text-shadow属性

向文本设置阴影。语法：

```
text-shadow: h-shadow v-shadow blur color;
```

> `text-shadow`属性向文本添加一个或多个阴影。该属性是逗号分隔的阴影列表，每个阴影有两个或三个长度值和一个可选的颜色值进行规定。省略的长度是0。

解释：

- h-shadow：必需。水平阴影的位置。允许负值。
- v-shadow：必需。垂直阴影的位置。允许负值。
- blur：可选。模糊的距离。
- color：可选。阴影的颜色。参阅CSS颜色值。

带有模糊效果的文本阴影

该例演示带有模糊效果的文本阴影。

电子出版　　　　　　移动阅读

HTML代码：

```
<p><span class="text-shadow-demo1">电子出版</span>   <span class="text-shadow-demo2">数字阅读</span>   <span class="text-shadow-demo3">移动阅读</span></p>
```

CSS代码：

```
span.text-shadow-demo1 {
    font-size: 1.35em!important;
    font-family: "黑体"!important;
    font-weight: bold;
    text-shadow: 0 5px 3px red;
}
span.text-shadow-demo2 {
    color: #fff;
    font-size: 1.35em!important;
    font-family: "黑体"!important;
    font-weight: bold;
    text-shadow: 1px 1px 3px #000;
}
span.text-shadow-demo3 {
    font-size: 1.35em!important;
    font-family: "黑体"!important;
    font-weight: bold;
    text-shadow: 0 0 3px #f75555;
}
```

word-break属性

规定自动换行的处理方法。

通过使用word-break属性，可以让浏览器实现在任意位置的换行。

语法：

```
word-break: normal|break-all|keep-all;
```

解释：

- normal：使用浏览器默认的换行规则。
- break-all：允许在单词内换行。
- keep-all：只能在半角空格或连字符处换行。

显示效果：

世上的武功确实没有高低之分，只有习武的人才有强弱之比，通过竞技，我们可以发现一个真正的自己，因为我们真正的对手，可能就是我们自己。The world of martial arts really no high and low points, only martial arts talent than the strength of the ratio, through the competition, we can find a real self, because we really opponents, may be our own.

上面的案例拆分了单词。HTML代码：

```
<p class="word-break-demo">世上的武功确实没有高低之分，只有习……
opponents, may be our own.</p>
```

CSS代码：

```
p.word-break-demo {
    width: 50%;
    word-break: break-all;
}
```

而在多看阅读器中拆分的单词，会自动添加连字符。

3.5 尺寸

CSS的尺寸分为三大类：相对、绝对、百分比。在制作过程中，推荐使用相对单位，而不是绝对单位。px、Pt、em之间换算表见9.5一节。

相对单位

CSS相对单位是相对于其他元素，如根元素、父元素来测量的。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸，任意浏览器的默认字体高都是16px。所有未经调整的浏览器都符合：1em＝16px。那么12px＝0.75em，10px＝0.625em。

比如在body元素中指定初始字体大小，比如`font-size:    16px;`，那么body的子元素就会根据这个来显示相应的尺寸。比如h1的字体大小设置成1.35em，则换算成px就是1.35×16＝21.6px。

rem是CSS3新增的一个相对单位（root   em，根em），区别在于使用rem为元素设定字体大小时，仍然是相对大小，但相对的只是HTML根元素。这个单位可谓集相对大小和绝对大小的优点于一身，通过它既可以做到只修改根元素就成比例地调整所有字体大小，又可以避免字体大小逐层复合的连锁反应。目前，除了IE8及更早版本外，所有浏览器均已支持rem。对于不支持它的浏览器，应对方法也很简单，就是多写一个绝对单位的声明。这些浏览器会忽略用rem设定的字体大小。一个例子：

```
html {
    font-size: 16px;
}
p {
    font-size: 1rem;
}
```

那么以后的都是1rem＝16px。px转em的在线网站——PXtoEM。

绝对单位

绝对单位的使用远小于相对单位，故在电子书制作中不推荐使用。

像素。图像显示的基本单位，译自英文「pixel」，pix是英语单词picture的常用简写，加上英语单词「元素」element，就得到pixel，故「像素」表示「画像元素」之意。像素换算成厘米的方法：（像素÷分辨率）×2.54。

英寸（in）。1英寸＝2.54厘米＝25.4毫米。在英制，12英寸为1英尺，36英寸为1码。

磅（Pt）。1英寸是72磅。例如，`font-size: 18pt;`等价于`font-size: 0.25in;`。

厘米。是十进制长度计算单位，符号cm。台湾称为「公分」。1厘米＝10毫米＝1/10分米＝1/100米＝0.394英寸。

毫米。是十进制长度计算单位，符号mm。台湾称为「公釐」。

百分比

CSS百分比是另一种指定相对大小的方法，给出与另一个值相关的值。百分比符号%。

3.6　书写规范

良好的书写规范，可以快速的帮助制作者理解样式含义，易于被阅读系统解析。更方便的是提高二次编辑时的效率。

书写顺序

CSS没有一个具体标准的书写顺序，一切以可维护性为准。在写大型的CSS样式时，最好遵循自己的一套标准，对做电子书来说也可以不遵循（样式很少）。本书样式表的顺序是用了一个工具美化的，见CSS属性在线排序。

如果实在记不住，可以按照英文字母的顺序依次书写。

使用CSS简写属性

CSS有些属性是可以缩写的，比如padding、margin、font等等，这样精简代码同时又能提高用户的阅读体验。新手刚学习CSS时，但不建议这样做。

去掉小数点前的「0」，比如0.85em就可以写成.85em。

16进制颜色代码缩写：有些颜色代码是可以缩写的，我们就尽量缩写，比如#ffffff就可以简写成#fff。

连字符、id

长名称或词组可以使用中横线来为选择器命名。

不建议使用「_」下划线来命名CSS选择器。因为输入的时候少按一个Shift键；浏览器兼容问题　(比如使用_tips的选择器命名，在IE6是无效的)。

不要随意使用id：class类选择器却可以重复使用，另外id的优先级class高，所以id应该按需使用，而不能滥用。

注释的写法

注释内容可以添加到样式名称前，也可以在样式结束的位置。示例：

```
/* 默认盒子独占居中100% */
div.picture-center {
  width: 100%;
  margin: 1.3em auto;
  text-align: center;
}
img.zhangyue-img {
  width: 100%;
}/* 默认图片宽100% */
```

另一种是，多个样式需要一起出现，它在设计时是属于一套组合样式，那么就可以采用下面的方面命名。

```
/* 篇章页（开始） */
样式内容区
/* 篇章页（结束） */
```

注意事项

下面的每条注意事项，建议是要遵守的。如果忘记，可能会带来一系列问题。最后排查时，比较费时间。

1. 样式声明一律小写；
2. 尽量用英文：把中文名字翻译为英文单词或中式英文句子，为了便于识别可以添加注释；
3. 样式命名尽量不缩写，除非一看就明白的单词。
4. 一旦选用或设定好命名规则，在编写时应保持一致格式。
5. 每条声明都建议用半角英文分号结束。

样式表命名

样式表命名也要遵循易读、有意义的规则来，比如：

字体样式表：Font.css、基础样式表：Stylesheet.css、打印样式表：Print.css等等。

如果一本书的样式表并不是很多，建议是合为一个样式文件。但常用的样式表可以分开单独存放，以方便重复使用。

第4章

正则表达式

正则表达式（Regular Expression）描述了一种字符串匹配的模式，可以用来检查一个串是否含有某种子串、将匹配的子串做替换或者从某个串中取出符合某个条件的子串等。换句话说，正则表达式就是记录文本规则的代码。

构造正则表达式的方法和创建数学表达式的方法一样。也就是用多种元字符与运算符可以将小的表达式结合在一起来创建更大的表达式。正则表达式的组件可以是单个的字符、字符集合、字符范围、字符间的选择或者所有这些组件的任意组合。

正则表达式是由普通字符（例如字符a到z）以及特殊字符（称为「元字符」）组成的文字模式。模式描述在搜索文本时要匹配的一个或多个字符串。正则表达式作为一个模板，将某个字符模式与所搜索的字符串进行匹配。

一般我们常用的是用字符串字面值（string literal）来匹配目标字符串的。所谓字符串字面值，就是字面上看起来是什么就是什么。比如你要在本书中搜索「元素」，只需要在查找框中输入「元素」即可。

「正则表达式是描述一组字符串特征的模式，用来匹配特定的字符串。」

——Ken Thompson

图形化工具

正则表达式的图形化工具，匹配内容：707-827-7019，显示效果如下：

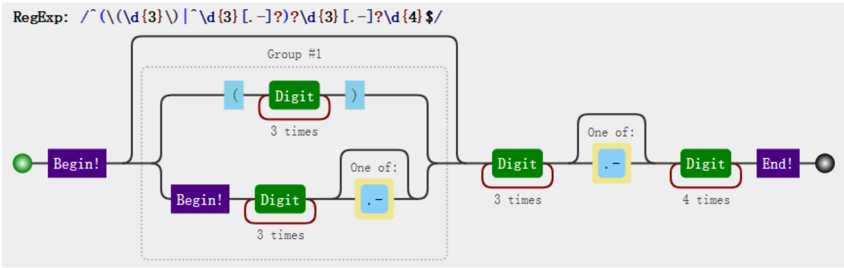


图4-1　正则表达式图形化

单词Digit的意思是数字，3 times意思是3次，one of的意思是之一，Group的意思是组。案例见4.5，可以看到工具能够更快的帮我们理清头绪。

用途

1. 匹配。判断给定的字符串是否与特定正则表达式相匹配；
2. 获取匹配的内容。使用正则表达式搜索指定的字符串，将符合条件的全部内容都提取出来；
3. 替换。使用正则表达式搜索指定的字符串，将符合条件的内容替换为特定的字符串；
4. 拆分字符串。

4.1 字符

正则表达式一般我会在两个软件中使用，一个是Emeditor，另一个就是Sigil。在Sigil中使用正则表达式的话，需要先了解这个软件的一些相关知识，如「查找&替换」的模式、选项配置。

在Sigil中使用正则表达式之前，你需要了解下5.3.16和5.4.17两节内容。

4.1.1 普通字符

正则表达式中有普通字符和元字符两种，普通字符是元字符之外的字符，元字符可以通过转义变成普通字符。

表4 正则表达式普通字符

符号	含义	举例	结果	备注
具体的字符	匹配字面值	picture	picture2 Picture	区分大小写
喝酒	要喝酒吗 我喝完酒了			

备注：加背景色为匹配结果，下文中匹配结果均是在Sigil中的。

4.1.2 特殊元字符

元字符为半角形式。

表5 正则表达式特殊元字符

符号	含义	举例	结果	备注
\n	换行符			由Enter键产生
\t	制表符			由Tab键产生
\s	任意一个空白字符			
\S	任意一个非空白字符			
\d	任意一个十进制数字			等价于[0-9]
\D	任意一个非十进制			等价于[^0-9]

	数字			
\w	匹配一个字母、数字、下划线、汉字	\w约翰	好人约翰 好人 约翰 好人：约翰 _约翰 John约翰 7约翰 （约翰）	等价于[a-zA-Z0-9_]
\W	匹配一个非字母、数字、下划线、汉字	\W约翰	好人约翰 好人 约翰 好人：约翰 _约翰 John约翰 7约翰 （约翰）	等价于[^a-zA-Z0-9_]

备注1：在Sigil中\w匹配不到汉字，在EmEditor中可以。

备注2：在EmEditor中，换行符为蓝色的↓，可以用\n匹配。

字母含义：数字（digit）\d、\D，字符（word）\w、\W，空白（space）\s、\S。

4.1.3　基本元字符

与、或、非、通配、转义等。左方括号「[」表示字符组起始，右方括号「]」表示字符组结束。左括号「(」表示分组起始，右括号「)」表示分组结束。

表6　正则表达式基本元字符

符号	含义	举例	结果	备注
.（英文句号）	匹配任意单个字符（除了换行符）	d.g	dog d.g d g dg drug	一个「.」代表一个字符。
喝..	我喝完酒了 大口喝，呛着了 大口喝。			
|	或	狗|猫粮	狗 狗粮 猫 猫粮 狗猫粮 狗粮和猫粮	左右分别作为一个整体对待，并不是只针对紧挨着的字符！ 左边的和右边的，出现哪个就匹配哪个，两个都出现就都匹配。
(狗|猫)粮 狗粮|猫粮	狗 狗粮 猫 猫粮 狗猫粮 狗粮和猫粮			

[]	匹配指定范围内的一个字符	r[aou]t	rat rot rut rt root	「[]」里无论有多少个字符，最终只输出一个 「[-]」一定得保证小的在左边！[9-0]是错误的。 「-」只有在「[]」里才是元字符，在其他地方都是普通字符，因此不需要转义。
喝[酒水]	要喝酒吗 喝酒水 自带酒水			
[-]	[0-9]个	0个 1个 9个 10个		
[A-Za-z]版	第B版 第b版 第fD版 第2版			
[^]	范围内的都不匹配，之外的任意一个字符都匹配	[^0-9]个	1个 10个 一个 n个 许多个	「^」作用于整个「[]」里内容！而不仅限于紧跟在「^」后面的内容。
\	对下一个字符转义 （把元字符转为普通字符；或者反过来）	d\.g	dog d.g d g	「\」自身也是元字符，如要匹配自身，需「\\」。
\n	匹配换行符	「n」是普通字符，通过「\」转义为特殊元字符。		

4.1.4　数量元字符

数量元字符用来指定正则表达式的一个给定范围必须要出现多少次才能满足匹配条件。左花括号「{」表示量词起始，右花括号「}」表示量词结束。

表7　正则表达式数量元字符

符号	含义	举例	结果	备注
*	匹配前一个字符（集合、子表达式）的0次或多次重复	图\d*	图2 图13 图1-3 图a 图中所示	贪婪型，总是尽可能多地匹配。
*?	*的懒惰型版本	图\d*?	图2 图13 图1-3 图a	遇到第一个符合匹配条件的就为止了。

+	匹配前一个字符（集合、子表达式）的一次或多次重复	图\d+	图中所示 图2 图13 图1-3 图a	贪婪型，总是尽可能多地匹配。
+?	+的懒惰型版本	图\d+?	图中所示 图2 图13 图1-3 图a	遇到第一个符合匹配条件的就为止了。
?	匹配前一个字符（集合、子表达式）的0次或一次重复	图\d?	图中所示 图2 图13 图1-3 图a	匹配某个字符（集合、子表达式），有它行，没它也行；当然如果有，也绝不放过它。
{数字}	匹配前一个字符（集合、子表达式）的几次重复	图\d{2}	图中所示 图2 图13 图1-3 图21 图123	前面的字符（集合、子表达式）必须连续出现指定的次数才算匹配。
{数字,数字}	匹配前一个字符（集合、子表达式）的几次到几次重复	图\d{2,4}	图2 图13 图135 图2150 图12367 图123-67	前面的字符（集合、子表达式）必须连续出现指定的次数范围才算匹配，并且总是尽可能多地匹配。前面的那个数字可以是0，所以{0,1}等价于?。
{数字,}	匹配前一个字符（集合、子表达式）至少重复几次	图\d{2,}	图2 图13 图135 图2150 图123-54	前面的字符（集合、子表达式）必须连续出现指定的次数或更多次数才算匹配，并且总是尽可能多地匹配。
{数字,}?	*的懒惰型版本	图\d{2,}?	图2 图13 图135 图2150 图123-54	遇到第一个符合匹配条件的就为止了。

如`.*`，它将会以贪婪的方式匹配主文本中的所有字符（数字）。贪心的量词，意味着它们在第一次尝试时会尽可能多地匹配字符。比如`7{1,}`和`7+`意义是一样的，另外`7{0,}`和`7*`、`7{0,1}`和`7?`也是。可以看出，花括号（或者说范围语法）是最灵活和精确的量词。

4.1.5 位置元字符

位置元字符起到精确定位作用。

表8 正则表达式位置元字符

符号	含义	举例	结果	备注
\b	匹配单词边界	\bcap	captain cap	\b在左边，只匹配开头

	（由字母、数字、下划线组成的，不管数量多少，都是一个单词）		recap recaps	
\b白色	白色 白色 工厂，白色 银白色 white白色			
cap\b	captain cap recap recaps	\b在右边，只匹配结尾		
\bcap\b	captain cap recap recaps	\b在两边，只匹配孤立的词		
\B	\b的反义	\Bcap	captain cap recap recaps	
		cap\B	captain cap recap recaps	
		\Bcap\B	captain cap recap recaps	
^	匹配字符串（段落）的开头	^When in ^白	（处在段首） When in the What and When in the （处在段首）白色的油漆很好 银白色的油漆也不错	
$	匹配字符串（段落）的结尾	book$ 书$	I have a book（处在段末） I have a book. 我有一本书（处在段末） 我只有一本书了	

备注：在EmEditor中，「\b白色」可以匹配到处于段首的「白色白色，引白色白色」。

4.1.6 回溯引用和前后查找

给正则表达式分组后就可以回溯引用了。前后查找也可以在编辑软件中使用如向前查找（查找前一个）、向后查找（查找后一个）等按钮实现，也可以使用正则表达式实现。

表9 回溯引用和前后查找

符号	含义	举例	结果	备注
()	子表达式	(ab){3}c	abababc	子表达式作为一个

(19|20)\d{2}	abbbc			整体对待。 子表达式里可以只有一个具体的字符，如(a)。 子表达式里可以有子表达式，即嵌套。
\数字	匹配第几个子表达式	H([1-3])H\1	H1H1 H2H2 H3H3 H1H2 H3H1	「\1」表示引用【查找内容】里的第一个子表达式。
(?＝)	向前查找	北京(?＝市区)	北京市区北京人口 北京的市区 北京市 北京人口总量 北京GDP总量 北京交通 上海市区	就是给子表达式加个前缀「?＝」，匹配时往前搜索，并且必须包含该子表达式，但输出结果不包含该子表达式。 「北京(?＝市区)」同「北京(市区)」，只是前者的输出结果不包含「市区」 向前查找模式可包含「.」和「＋」之类的元字符
(?!)	负向前查找	北京(?!市区)	北京市区北京人口 北京的市区 北京市 北京人口总量 北京GDP总量 北京交通 上海市区	就是给子表达式加个前缀「?!」，匹配时往前搜索，只要不包含该子表达式就算匹配。
(?＜＝)	向后查找	(?＜＝市区)面积	北京市区面积很大 北京市区的面积 北京的土地面积	就是给子表达式加个前缀「?＜＝」，匹配时往后搜索且必须包含该子表达式，但输出结果不包含该子表达式！ 「(?＜＝市区)面积」同「(市区)面积」，只是前者的输出结果不包含「市区」。 向后查找模式不能包含「.」和「＋」之类的元字符。
(?＜!)	负向后查找	(?＜!市区)面积	北京市区面积很大 北京市区的面积 北京的土地面积	就是给子表达式加个前缀「?＜!」，匹配时往后搜索，只要不包含该子表达式就算匹配。
(?(数字))	如果…，那么…	(\d)(A)?(?(2)B)	8AB 8A 3BB 2	如果第几个子表达式匹配成功，那么继续匹配后面的；否则，后面的就当

				不存在。
(?(数字)|)	如果不…，那么…	(\d)(A)?(?(2)|B)	8AB 8A 3BB 2	如果第几个子表达式匹配不成功，那么继续匹配后面的；否则，后面的就当不存在。

本节主要内容来自《InDesign/Office印前排版综合教程》一书的「附录B 正则表达式」，该书ISBN为978-7-121-29647-5。感谢！

4.2　转义与回溯引用

正则表达式的使用需要注意有两种规则：一是转义，即查找元字符本身；二是分组，以进行回溯引用。

转义

如果你想查找元字符本身的话（在正则表达式搜索模式下），比如你查找.或者*就出现了问题：你没办法指定它们，因为它们会被解释成别的意思。要匹配这些特殊字符，必须首先使字符「转义」，即将反斜杠字符\放在它们前面。

在Sigil软件中，你可以设置「查找&替换」面板中的「选项」，勾选「自动转义」，为需要转义的字符自动添加转义符（需配合Ctrl＋F使用）。

分组与回溯引用

分组是正则中非常强大的一个功能，可以让上面提到的量词作用于一组字符，而非单个字符，分组的语法是圆括号包裹(xxx)。

```
(abc){2} //  匹配abcabc
```

使用小括号指定一个子表达式后，匹配这个子表达式的文本（也就是此分组捕获的内容）可以在表达式或其它程序中作进一步的处理。默认情况下，每个分组会自动拥有一个组号，规则是：从左向右，以分组的左括号为标志，第一个出现的分组的组号为1，第二个为2，以此类推。每个缓冲区都可以使用「\n」访问，其中n为一个标识特定缓冲区的十进制数。

回溯引用用于重复搜索前面某个分组匹配的文本，这个主要用在替换上。比如对标题的格式进行替换，待匹配标题格式如下：

```
第一章　侯门庶子
```

首先我们先分析下这个标题格式，「第一章」如果全文有三十章或更多，比如这样「第三十章」、「第一百一十章」等等，我们发现「第」与「章」这是一个固定的字符，变化的只是两个字符之间的章节编号，字符数量从1（一）到4（一百一十）。然后是章节标题内容，如果都是统一的4个字最好办，如果不是，那只需限定最少字符数量。那么，正则表达式书写如下：

```
查找：<h1>第([\x{4e00}-\x{9fa5}]{1,})章 ([\x{4e00}-\x{9fa5}]{4})</h1>
替换：<h2 title="第\1章 \2"></h2><p class="chapter-title">第\1章</p><p
class="chapter-subtitle">— \2 —</p>
```

`([\x{4e00}-\x{9fa5}])`这串字符表示的Unicode编码中的「中日韩统一表意文字」。

4.3 字符组

正则表达式除了特殊元字符之外，还有一些字符集：

- ([\x{4e00}-\x{9fff}])：中日韩认同表意文字区
- ([\x{2e80}-\x{fe4f}])：匹配任意CJK字符
- ([0-9])：匹配全部半角数字集合（0,1,2,3,4,5,6,7,8,9）中的任何1个，等价于\d，等价于[0123456789]
- ([a-z])：匹配全部小写半角字符的集合（a,b,c……x,y,z）中的任何1个
- ([A-Z])：匹配全部大写半角字符的集合（A,B,C……X,Y,Z）中的任何1个

在EmEditor中匹配汉字的字符集是`(\u4e00-\u9fff)`。

比如目前小说类的标题有以下几种格式，分别使用正则表达式替换修改。

格式1：

```
第一章 标题内容
第一百一十一章 标题内容（一）
查找：第([\x{2e80}-\x{fe4f}]{1,})章 (.*?)
替换：第\1章 \2
```

格式2：

```
第一章：标题内容
第一百一十一章：标题内容（一）
查找：第([\x{2e80}-\x{fe4f}]{1,})章：(.*?)
替换：第\1章：\2
```

格式3：

```
0001.标题内容
0001 标题内容（一）
查找：([0-9]{1,})\.(.*?)
替换：\1.\2\n
```

```
查找：([0-9]{1,}) (.*?)
替换：\1 \2\n
```

格式4：

```
第1章标题内容
第111章标题内容（一）
查找：第([0-9]{1,})章(.*?)
替换：第\1章 \2
```

以上正则使用时可根据实际情况增加<p></p>或<h1></h1>等，另有节/篇/部/卷/条等。

正则表达式并不需要刻意去记，只要理解上面的内容，就可以根据实际的情况自己书写正则表达式。然后是在制作ePub电子书时，首先书写一个表达式，验证成功后再全文替换。

下面的案例演示下如何把正文标题填充到网页标题（<title></title>）中，前置工作是格式化了代码。

<title>(.*?)</title>\s+<link hre(.*?)>\s+</head>\s+<body>\s+<h1>(.*?)</h1>

\1　　\2　　\3

<title>\3</title><link hre\2></head><body><h1>\3</h1>

图4-2　修改＜title＞的内容

HTML代码：

```
<title>ePub入门</title>
  <link href="../Styles/Stylesheet.css" type="text/css" rel="stylesheet"/>
</head>

<body>
  <h1>1 ePub概述</h1>
```

查找替换的正则表达式：

```
查找：<title>(.*?)</title>\s+<link hre(.*?)>\s+</head>\s+<body>\s+<h1>(.*?)</h1>
替换：<title>\3</title><link hre\2></head><body><h1>\3</h1>
```

替换之前检查下「模式」、「选项」是否设置正确；正则表达式写完之后，先进行一个项的替换，看实际的替换结果是否正确，若不正确，则

修改。然后根据标题层级自行更换标题标签。

也可以参照下面的例子：

```
<!--原文本-->
<title></title>
</head>
<body>
<h1>XXXX</h1>
<!--目标文本-->
<title>XXXX</title>
</head>
<body>
<h1>XXXX</h1>
```

正则：

```
查找：(?s)<title></title>\n(.*<h1>(.*)</h1>)
替换：<title>\2</title>\r\1
```

在写正则表达式之前先对要匹配的文本进行分析，哪些是变化的，这个变化的部分可以用哪些正则表示。不变的地方，为了缩减正则表达式的长度可以适当用`(.*?)`代替，然后找出文本的规律。比如像之前的几个案例，「第x章 标题内容」，x就是变化内容，「标题内容」则有字数或其他规律，如果你想省事可以直接用`(.*?)`代替，但是在使用时需要注意。

然后是在Sigil中使用时就需要补上匹配文本前后的HTML标签就行了。标签起到了限定匹配范围的作用（与^和$类似）。

在ePub电子书的制作中，正则表达式一方面可以查找文内错误，比如空格、断行、标点大小写、标点全角半角转换（需要配合搜索模板使用）等；另一方面可以批量对有规律的标题、斜体书名、诗歌添加样式等。这里的有规律指的是字数固定、格式固定等。

还有一种情况是，原始资源已经是ePub的，我们想要再进行精加工，比如各家对注释规范的写法不同，我们就可以用到正则表达式来完成批量修改。

4.4　应用

上面讲了一些常用的正则表达式字符组，那么我们就可以根据这些字符组进行组合，去匹配文中的某些规律文本。比如在制作ePub电子书时，有些字与字之间的空格需要删除、半角符号转全角、非标准换行等。

断行

断行又称作非标准换行，即一句话没有说完而新起了一段。这在格式转换中经常发生，比如图像式PDF经过OCR（光学字符识别，Optical Character Recognition，OCR。指对文本资料的图像文件进行分析识别处理，获取文字及版面信息的过程。）之后就存在大量的断行，复制文本PDF中的文字也会保留断行信息。

那么查找的正则表达式就可以这么写：

Sigil中的规律——「字＋</p>」，正则表达式则为：

```
查找：([\x{4e00}-\x{9fa5}])</p>\s+<p>
替换：\1
```

Emeditor中的规律——「字＋换行符」，正则表达式则为（\n指换行符）：

```
查找：([\u4e00-\u9fa5])\n
替换：\1
```

断行的查找只能人工一个个排查，因为有些标题也有标点符号。替换的正则表达式根据回溯引用的规则就可以自己写出来。

还有一种正则表达式也可以查找断行（在Sigil中）：

```
查找：([^。>\?？\."」！……：!]|[^。>\?？\."」！……：!])</p>\s+<p>
替换：\1
```

样式修改

比如我想把一个普通段落格式化成标题：

```
<p>第一章 侯门庶子</p>
```

正则表达式：

```
查找：<p>第([\x{4e00}-\x{9fa5}]{1,})章 (.*?)</p>
替换：<h1>第\1章 \2</h1>
```

如果<h1>要添加class的话，也可以在替换时输入进去。

引号匹配

有的书是半角引号，这样在转成全角时只能替换成一种引号，前双引号或后双引号。这种情况就需要用正则表达式查找，然后进行匹配替换。

```
查找：(「)+([^」「]*)「
替换：\1\2」
查找：(\n[^「」\n](?=<!「)[^「」\n]」|「[^「」\n]+(?=「)|「[^」「\n]+
\n(?!」))+
```

清理HTML样式或其他属性

有时我们需要用Word转出来的HTML文件，针对这种情况，通常的建议是先删除标签内的属性信息，以保证内容文档的干净。正则表达式：

```
查找：<p.[^<>]*>
替换：<p>
```

<p>标签可以根据实际情况，进行更改，比如table、th、td、span等。

在制作中很少有一次正则可以处理的，需要多次正则才可以完成一个内容的修改。这种情况下，需要特别注意处理的先后顺序。

英文斜体书名

例如：

> 本章内容是《Sigil用户指南》（Sigil User Guide）的中文版本。

替换的样式可以是自己写的行内斜体样式或者是斜体标签。

```
查找：》（([a-z A-Z]*)）
替换：》（<i>\1</i>）
```

批量替换英文标点

批量替换英文逗号

```
查找：([^A-Za-z<"0-9,]),+([^A-Za-z",0-9])
替换：\1，\2
```

批量替换英文句号

```
查找：([^A-Za-z<"0-9.])\.([^A-Za-z"0-9.])
替换：\1。\2
```

批量替换英文分号

```
查找：([^A-Za-z<"0-9;]);+([^A-Za-z"0-9;])
替换：\1；\2
```

批量替换英文感叹号

```
查找：([^A-Za-z<>"0-9])\!([^A-Za-z"0-9])
替换：\1！\2
```

批量替换英文问号

```
查找：([^A-Za-z<>"0-9])\?([^A-Za-z"0-9])
替换：\1？\2
```

批量替换英文冒号

```
查找：([^A-Za-z<>"0-9])\:+([^A-Za-z"0-9])
替换：\1：\2
```

可以把每次的搜索保存成一个搜索模板，使用时再直接调用。

参考如何查找硬回行以及批量修改英文标点。

删除空格

各种标点符号前后的空格。这些根据半角全角标点符号前后、「正常」查找替换模式、「正则表达式」模式下，从而进行了细分。然后就是在书写删除空格的正则表达式时，要判断这些空格是不是可以直接删除还是需要人工判断。可以直接删除的编到一组，需要人工判断的编到另外一组。

Sigil中有一个「搜索模板」功能是专为「查找&替换」而设计的，它可以保存搜索条目，以便后续使用时可以直接调用。具体使用方法见5.3.17节。

内嵌双引号改单引号

这个需求是群里 @知秋 在处理一本小说时，需要解决的问题。具体情

况是：双引号里面也是双引号，里面的双引号替换成单引号。如：

```
「它一直在高速前进，按照前进的路线方向……它应该正在朝「九蝉神殿」飞去。」波林说道。
```

群里 @侠隐无名 提供了两个正则：`「「([^「」p]+)」([^「」p]+)」`和`「([^「」p]+)「([^「」p]+)」)」`。这两个正则仍有问题，这里修改如下：

```
查找：「([^「」]*?)「([^「」]+)」([^「」]*?)」
替换：「\1‘\2’\3」
```

备忘，记录在此处。

常见的正则表达式

Email地址：`^\w+([-+.]\w+)@\w+([-.]\w+).\w+([-.]\w+)*$`

域名：`[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(/.[a-zA-Z0-9][-a-zA-Z0-9]{0,62})+/.?`

InternetURL：`[a-zA-z]+://[^\s]` 或 `^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=])?$`

手机号码：`^(13[0-9]|14[5|7]|15[0|1|2|3|5|6|7|8|9]|18[0|1|2|3|5|6|7|8|9])\d{8}$`

电话号码("XXX-XXXXXXX"、"XXXX-XXXXXXXX"、"XXX-XXXXXXX"、"XXX-XXXXXXXX"、"XXXXXXX"和"XXXXXXXX)：`^((\d{3,4}-)|\d{3.4}-)?\d{7,8}$`

国内电话号码（0511-4405222、021-87888822）：`\d{3}-\d{8}|\d{4}-\d{7}`

身份证号（15位、18位数字）：`^\d{15}|\d{18}$`

短身份证号码（数字、字母x结尾）：`^([0-9]){7,18}(x|X)?$` 或 `^\d{8,18}|[0-9x]{8,18}|[0-9X]{8,18}?$`

日期格式（XXXX-XX-XX）：`^\d{4}-\d{1,2}-\d{1,2}`

4.5 阅读正则表达式

下面用一个例子来加深对正则表达式的认识。

```
正则表达式：(\d{3,4}[.-]?)+
匹配内容：707-827-7019
```

加号表示出现一次或多次。这个正则表达式表示括号里的模式出现一次或多次，括号里的模式匹配三位或四位数字，后跟一个连字符或一个点号。

下面逐一解释表达式中的每一项：

- 左圆括号(为捕获分组的起始符；
- 反斜杠\为字符组简写式的起始符（对之后的字符进行转义）；
- 字符d为字符组简写式的结束符（d匹配0到9范围内的任意数字）；
- 左花括号{为量词起始符；
- 数字3为匹配的最小数量；
- 逗号,隔开不同的数量；
- 数字4为匹配的最大数量；
- 右花括号}为量词的结束符；
- 左方括号[为字符组的起始符；
- 点号.（匹配点号本身）；
- 连字符-匹配连字符的本身；
- 右方括号]为字符组的结束符；
- 问号?表示量词「零个或一个」；
- 右圆括号)为捕获分组的结束符；
- 加号+表示量词「一个或多个」。

这个表达式能用但不完全对，因为它只能匹配3位或4位的数字，而不管是否符合电话号码的格式。我们来改进一下：

```
(\d{3}[.-]?){2}\d{4}
```

这个表达式匹配的字符串是连续两个无括号的三位数字，每三位数字后可以带连字符也可以不带，最后是一个四位数字。最后这个正则表达式表示第一个3位数字可以带也可以不带括号，即区号是可选的：

```
^(\(\d{3}\)|^\d{3}[.-]?)?\d{3}[.-]?\d{4}$
```

为了便于理解，我们再依次看一下表达式中的各项：

- 出现在正则表达式起始位置或者竖线（|）之后的脱字符^，表示电话号码会出现在一行的起始位置；
- 左括号(为捕获分组的起始符；
- \(表示左括号本身；
- \d 匹配一位数字；
- \d之后的{3}是量词，表示匹配三位数字；
- \)匹配右括号本身；
- 竖线符|表示选择，也就是从多个可选项中选择一个，换句话说，它表示「匹配一个不带括号的区号或一个带括号的区号」；
- 脱字符^匹配行起始位置；
- \d 匹配一位数字；
- {3}是表示匹配三位数字的量词；
- [.-]?匹配一个可选的点号或连字符；
- 右括号)为捕获分组的结束符；
- 问号?表示分组可选，即分组中的前缀可有可无；
- \d 匹配一位数字；
- {3}表示匹配三位数字的量词；
- [.-]?匹配另一个可选的点号或连字符；
- \d 匹配一位数字；
- {4}是表示匹配四位数字的量词；
- 美元符$匹配行结束位置。

这个表达式最终匹配十位的北美电话号码，而且括号、连字符或者点号都是可选的。你可以试试不同格式的电话号码，看看它能否匹配。

第5章

ePub编辑——Sigil

使用Sigil可以所见即所得的编辑ePub电子书，Sigil的Github链接。本章内容是Sigil官方用户指南的中文版本，部分教程是赤霓原创。英文版本为0.7.2版，译者所用Sigil的版本是0.9.8，所以与0.7.2版不同的地方以0.9.8版为准，也适当地增加了在0.7.2版本后新增的功能描述。

版权归Dave Heiland所有。

5.1 介绍——Sigil：ePub编辑器

欢迎来到Sigil用户指南——Sigil 0.9.8版。

Sigil 0.9.8是志愿者经过无数小时努力的结果，创造的最快、最稳定、最强大、版本最好的免费ePub编辑软件。

跳转到教程，一步步了解如何使用Sigil。或查看最新版本更新的功能。

下载Sigil：

- Sigil官方网站

在MobileRead中Sigil获得了用户和开发人员的帮助：

- Sigil Forum @ MobileRead

「Sigil用户指南」将帮助你更好地编辑ePub的文档，其中包括：

- 安装：在Windows、MAC和Linux上安装Sigil。
- 功能：完整地描述所有Sigil的功能。
- 教程：一步步指导您如何使用Sigil创建书籍。
- 常见问题：有关Sigil的常见问题。
- 为**Sigil**做贡献：如何支持Sigil。

> 您可以下载ePub格式的Sigil用户指南或免费阅读。

5.1.1 更新

Sigil的最新版本包含数百个更改、BUG修复和新功能，同时仍保留与早期版本相同的易于操作的界面。几乎所有的Sigil的代码都已更新和改进，特别关注的是以下几个方面：

- 稳定性：所有已知的崩溃原因和以前版本的相关错误都被识别并修复——这使Sigil可以稳定更新您的ePub。
- 鲁棒性：Sigil可以在阅读文件时修复常见的问题，并且不会在打开无效ePub时退出程

序。增强了保存时以执行更多的完整性检查。现在您可以在任何地方完成拆分和合并，而没有潜在的数据丢失状况。切换视图时，光标位置正确更新，并且通过一个或所有文件一致而准确地查找和替换工作。

- 性能：Sigil在许多方面的操作更加顺畅，如打开和切换标签、生成目录、合并文件、批量重命名、替换文本等。
- 灵活性：添加了更多的首选项选项和功能，让您以所需的方式使用Sigil。

> 有关对Sigil所做更改的详细列表，请参阅更新日志。

5.1.2 关于Sigil

Sigil旨在通过以ePub格式来轻松设计出色的电子书。如果您正在为自己的书籍进行排版，或者您是在多个平台上出版书籍的专业编辑人员，那么Sigil就是为您而准备的。您可以使用Sigil排版书籍，然后打包成一个ePub电子书，它完全符合您的预期效果。另外高级功能的使用，让Sigil成为最受欢迎的ePub编辑器之一。

这个免费且开源的软件完全由志愿者编写和支持。Sigil的维护人有Kevin Hendricks和Doug Massay，John Schember是以前的维护人员。Sigil最初是由Strahinja Markovic（已隐退）于2009年创建的，完全免费下载和使用。其他志愿者还提供支持，他们为用户的问题提供代码、翻译和解答。有关贡献者的完整列表，请参阅Sigil中菜单栏的「帮助→关于」。

下面的内容来自Sigil官网的About页面：

什么是Sigil？

Sigil是一个多平台的EPUB电子书编辑器（类似Word或LibreOffice，但是专门用于EPUB电子书），具有以下特点：

- GPLv3下的免费和开源软件
- 多平台：在Windows、Mac和Linux上运行（注意：Linux仅获得有限的支持）
- 完整的UTF-16支持
- EPUB 2规格支持与有限的EPUB 3支持
- 多个视图：书籍视图、代码视图和预览视图
- 「书籍视图」中的所见即所得编辑
- 在「代码视图」中完成对直接编辑EPUB语法的控制
- 目录生成器与多级标题支持
- 元数据编辑器
- 翻译成多种语言的用户界面
- 使用默认字典和用户配置字典进行拼写检查

- 为查找和替换服务的完整正则表达式（PCRE）支持
- 支持导入EPUB和HTML文件、图像和样式表，文件可以通过集成的EPUB验证器进行验证
- 所有导入的文件格式都将更正，并且可以选择清除您的编辑
- 插件！

历史

这是无聊的部分，所以这就是为什么它之后的功能是人们真正关心的。其中大部分来自记忆，也通过电子邮件进行回顾。Sigil已经存在很多年了，在那段时间里它几经易手。

开始

Sigil由Strahinja Marković于2009年创建。这是他作为在校计算机科学课程项目工作中的一部分，是他学士和硕士论文的基础。由于这是一个在上学期间负责的项目，因此有足够的时间致力于Sigil的开发工作。一旦Strahinja大学毕业，再保持像Sigil这样项目的规模就不那么简单了。

简单地说，他放弃Sigil有几个原因：1）他得到了一份工作，这份工作占用了他大部分的空闲时间。2）在同一个代码上日复一日地工作了两年之后，他厌倦了。他的原话是「我已经研究这个代码库两年多了，坦率地说，我已经厌倦了它。」。3）他觉得是时候继续前进并贡献于其他项目，并以更自由的方式利用空闲时间，比如开展其他开源项目。

Strahinja离开Sigil和新的开始

2011年6月，Strahinja在当时的Sigil开发博客上写了一篇文章，指出他要离开，需要一名新的维护人员。

我（John Schember）曾为calibre开发过产品，后来决定提出要成为Sigil的维护者。到2011年7月，我是代码库的正式维护者。

Strahinja在过渡期间对我说了一些非常友好的话。他对我的信任使我觉得我适合这个角色。具体来说，他说的这两件事一直困扰着我（作为开发人员我非常尊重他）。

1. 「坦率地说，即使有人提出愿意比你花更多的时间在Sigil上，我仍然可能让你担任PPL/gatekeeper的角色，因为我知道我可以相信你的判断。其他人则可以在您认为必要的任何级别的监督下作出贡献。」
2. 「就像我说过的，我不打算制定任何期望/要求。如果你决定以维护者的身份接管这个项目，你就可以自由支配Sigil开发的任何方面；你已经成为了「老板」，如何评价他人（和自己）未来的贡献取决于你自己。我不会以任何方式干涉。我真的希望你知道你有充分的自主权去做任何你认为必要的事情。我相信你的判断。」

让其他人完全控制你的项目，一个你开始的项目，基本上对该项目的独立开发者是非常困难的。他基本上把他的经典野马车的钥匙交给了我，

说它是你的，你想干什么就干什么，即使你弄坏了它我也不介意。

期望和现实

当我（John　Schember）接手时，我是计划成为Sigil的项目经理。我的目标是管理发行版、协助贡献者、评估代码，而不是自己编写代码。我对维护Sigil的目标是从项目经理的角度做出的，并继续为calibre编写代码。

与calibre相比，只有很少的贡献者愿意提供帮助，这对Sigil来说一直是个问题。在接手Sigil时，0.4.0版本已接近就绪。只有几个模块不见了。在了解C＋＋和Qt（Sigil是用哪一个编写的）之后，我决定完成0.4.0版本，并把它发布出去。

由于贡献者寥寥无几，最终我几乎全职完成了绝大多数开发工作（就像Strahinja所做的那样）。我需要休息一下，在2013年末，我干脆停止了编写代码。这些年来有几个贡献者，但他们总共只有3-4个人，他们都想做一些具体的事情。那很好，非常自然也可以接受。贡献者可以做他们想要做的任何事情，这就是开源软件的意义所在。

真的有两个人帮助完成了0.7.x版本，一旦0.7.4版本发布，两个贡献者就觉得他们已经完成了他们想要完成的任务，然后退出Sigil这个项目。那时我说我无法继续保持目前的开发进度，我需要帮助。没有任何帮助，所以我休息了一下。

Sigil并没有停止开发，我还不时地开发它，但不多而且变动很轻。就像Strahinja一样，生活发生了变化，更重要的事情发生了。再加上像他这么长时间的工作，我觉得有点累坏了。0.7.4版于2013年10月27日发布。

Sigil的Spiritual Successor

当我说我不能像Sigil的唯一开发者那样工作时，有人表现得像天塌下来一样。来自calibre的Kovid　Goyal成为了Sigil的Spiritual　Successor，他是calibre的EPUB编辑。我没有提供任何代码来完成这项工作，但是与Kovid讨论了一些功能。他发布了它，我于2014年2月9日宣布：「此时Sigil不再积极开发。」自从0.7.4发布以来，我一直没有接触Sigil的代码。但至少有一些开源的、稳定的，并有一个人们可以使用的积极开发社区。

很多人认为没有积极开发意味着Sigil死了。我没有。我只是没有贡献任何代码。它还在那里，只是在冬眠，等待人们作出贡献。

2014年9月

2014年9月，calibre的编辑担任了一年多的主要开发，拥有了绝大多数功能，但很多人不喜欢它与calibre联系在一起的事实。大多数人不喜欢他们必须安装完整calibre只是为了使用编辑器。

Kevin Hendricks决定出面，说他已经受够了，希望看到Sigil继续下去。从那时起，他一直是编写代码的主要力量。他甚至在我时间允许的情况下把我叫回来。在他的帮助下，Sigil已经发布了多个新版本，并且开始了良好的发展。

2015年6月

维护人员从John Schember转到Kevin Hendricks和Doug Massay。John离开的原因很简单。1）缺乏时间是主要原因。2）缺乏动力。这不再有趣了。

大多数人不知道的是John从来没有真正使用过Sigil。除了这是一个有趣的工作项目。他的动机是要有一个免费的开源电子书编辑器可供需要它的人使用。唉，时代和兴趣都在变化，和以前一样，最好把Sigil交给那些（只要他们愿意的话）继续让它变得伟大的人，而不是让它死去。

遗漏的东西

开次，Sigil是在GoogleCode上托管的，它有一个谷歌群组（GoogleGroup），甚至有一个Twitter订阅源。由于GoodCode基本上处于关闭状态，因此开发工作于2013年9月转移到GitHub。拥有多个支持论坛没有意义，所以全部的内容都整合到了MobileRead中，我不喜欢或使用Twitter，所以它就停止了。我认为没有必要有一个长格式的博客，有一个RSS订阅源来通知人们发布的内容，还有一个Twitter帐户来指向博客就够了。

5.1.3 关于ePub

ePub（电子出版物）格式是最流行的电子书格式之一。这是一个可供任何人使用的、免费的开放标准。

ePub电子书是一个使用普通zip格式，其他几个文件存储在一起的文件集合。这些文件包括文本、图像、目录、样式表、字体和关于书籍的详细信息，如标题、作者等。ePub这个标准格式，意味着您的电子书可以在许多电子阅读器上阅读，或者轻松转换为其他电子书阅读器支持的格式。

您的ePub文件使用HTML格式（超文本标记语言——用于显示文本和图像的一种代码）将您的文本存储在网页中。这表示HTML只包含内容文

本和标记标签，如`<h1>`或`<p>`，它告诉阅读器如何格式化和显示文本——可以是粗体、斜体、普通段落、标题等。

除了用于控制或样式设计的HTML标签之外，ePub书籍通常使用样式表来格式化HTML文件。样式表是ePub中的另一个文件，其中包含使用HTML标签来显示任何文本的样式（浏览器默认样式）。如`<h1>`（一级标题）应该更大或居中。这样，您可以将书籍的内容与其样式分离，并方便您轻松地调整ePub的样式。

当然，当有人阅读ePub时，他们看不到所有HTML文件或样式表。电子阅读器只是显示文本和图像，就像一本正常的书——尽可能接近您设计的格式，即使这个文件有数千页。

开始使用Sigil时，您不必了解ePub格式的所有细节，或了解HTML代码。然而，如果您已经花了很多时间使用Sigil，您必然需要熟悉这些知识，以充分利用Sigil的高级功能，最终实现您想要的设计效果。

> 文本文件实际上以XHTML格式（HTML兼容版本的HTML）存储，但是为了使指南更容易阅读，我们将使用HTML来引用HTML或XHTML。

Sigil支持ePub 2，也支持ePub 3中的一些功能。

> 即使Sigil为制作ePub文件而设计，ePub格式也可以轻松转换为其他电子书格式，如mobi。

要了解有关HTML的更多信息，您可以阅读各种在线指南，其中包括：

- W3Schools HTML Guide

可通过MobileRead的用户获得关于ePub格式的帮助：

- ePub Forum @ MobileRead

如果您真的想了解开发Sigil的官方ePub标准的细节，您可以阅读原始技术规范：

- Official ePub 2.0.1 IDPF Standard
    - **OPS**（开放出版物结构）：描述ePub文件中的所有内容的关键规范。
    - **OPF**（打开包装格式）：ePub文件如何相互关联，元数据详细信息（content.opf）。
    - **OCF**（**OEBPS**容器格式）：如何将文件打包成.epub文件。

> 您可以下载ePub版本的Sigil用户指南（*Sigil_User_Guide*），该指南使用Sigil中创建，也可以在Sigil中查看，您还可以查看ePub电子书的示例。

5.2 安装——入门

您可以从Sigil官方网站下载Sigil的最新版本，并将其安装在您的操作系统中。

Windows

Sigil支持Windows 7，也可以在Windows XP、Vista、Windows 7或8以上系统运行。

- Download Sigil for Windows

双击文件以运行安装程序，例如：

- Sigil-0.9.x-Windows-Setup.exe

Mac

Sigil支持MAC OS 10.8（Mountain Lion）x64位Intel处理器，但应该运行在10.7或更高版本中。

- Download Sigil for MAC

打开DMG文件并将「Sigil.app」文件拖到您的应用程序文件夹，例如：

- Sigil-0.9.x-Mac-Package.dmg

Linux

由于Linux的版本数量，Sigil的二进制版本不可用。

然而，由于源代码是免费的，您可以用源代码创建您的Linux版本：

- Building Sigil From Source Code

5.3 功能——深入指南

本节快速概述了所有Sigil的功能，然后在后续章节中详细介绍了Sigil的具体内容。

您还可以阅读教程部分，了解在Sigil中常见任务的操作说明，然后再返回到此处了解更多详细信息。

编辑

您可以使用「书籍视图」或「代码视图」来编辑您的书籍。「书籍视图」编辑器是一个所见即所得的编辑器，可以显示您的图书预览效果，并且在需要进行基本的样式更改时很有用。当您想要完全控制您的书籍或您想使用Sigil的高级功能时，「代码视图」编辑器允许您直接编辑HTML代码。您还可以使用带有「代码视图」的实时「预览」窗口来显示当前页面的预览效果。

查找&替换

搜索或替换文本的能力是Sigil最强大的功能之一。「书籍视图」提供了一个基本的查找功能，但在「代码视图」你可以使用「查找&替换」的全部功能。您可以在一个HTML文件或所有HTML文件中搜索或者替换文本，您可以只搜索字面值或者使用正则表达式修改文本、格式。您还可以使用「搜索模板」来使用保存的搜索。

管理文件

「图书浏览器」按文件类型来组织ePub中的所有文件，并且支持查看或修改它们。双击一个文件，在编辑窗口中打开它，或右键单击所选文件，进行查看、重命名、删除、合并、添加、链接样式表等操作。HTML文件按照纸书中的顺序列出——您可以拖动它们来重新排序。您也可以轻松地将文件添加到您的书中。

目录

「目录」（TOC）窗口将显示书籍的目录是什么样的，并允许您通过单击章节名称跳转到具体章节。您还可以随时根据书籍中的标题标签生成新的目录、编辑目录或创建一个HTML版本的目录文件。

修改图书详细信息

「元数据编辑器」提供了更新标题、作者和语言的快捷方式。您可以添加数百种ePub标准元数据条目中的任何一个，如出版日期、合著者、版权信息等。

首选项

在「首选项」窗口中可以自定义Sigil中的许多设置。特别是您可以设置Sigil的用户界面语言、键盘快捷键、外观、字典等。

验证

Sigil包括FlightCrew ePub验证工具（0.9.8的版本已剔除，使用此功能需安装插件），以检查您的书是否符合ePub标准，并且可以通过在线W3C CSS验证服务验证您的CSS样式表。Sigil还可以检测您的代码，在打开时是不是合规的XHTML文件，并且还可以使用HTML Tidy自动清理或格式化代码。

报告

可以生成报告以显示相关文件的详细信息，包括文件大小、字数、图像是否在使用、使用了哪些样式、书中的字符等。

拼写检查

您可以使用强大的「拼写检查」对话框检查文本的拼写是否正确，可以查看所有拼写错误的单词，并将其保存到字典中或立即修改它们。可以创建多个用户字典，并可以同时使用一个或多个字典。

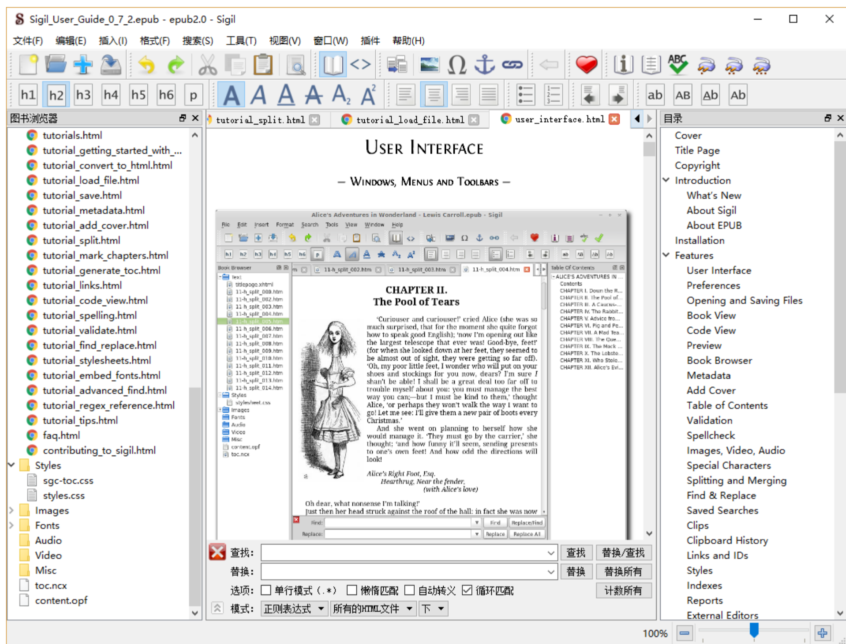
高级工具

Sigil有许多自定义工具来帮助编辑您的电子书——包括「搜索模板」（保存查找替换搜索）以及「剪切板编辑器」（粘贴文本或样式）。可以自动删除未使用的媒体文件和样式，可以使用外部程序编辑文件，也可以直接跳转到HTML的样式所在样式表中的位置。

语言支持

Sigil已被翻译成超过25种语言，可以根据需要轻松更改用户界面语言。它以C＋＋编写，用于高性能的需求并且包括完整的UTF-16支持。

5.3.1 用户界面：窗口、菜单和工具栏



Sigil的用户界面包括：

- 窗口：显示信息，如书籍视图、图书浏览器和目录。
- 主菜单：这里列出了大部分命令，如「文件→打开」。
- 工具栏按钮：可以拖动按钮组，也可以快速访问许多菜单项。
- 键盘快捷键：所有菜单项的用户自定义快捷键，如Ctrl＋S保存文件。
- 右键菜单：右键单击某个文件或窗口时显示的命令。

> 用户指南中的屏幕截图可能与您在当前计算机上看到的界面略有不同。

窗口

Sigil使用主界面中的各种窗口来显示信息：显示文件的窗口（图书浏览器）、书籍编辑、查看目录（目录）等。

窗口可以隐藏——使用「视图」中的菜单条目来显示或隐藏指定的窗口。

某些窗口也可以移动。要移动窗口，请单击其标题并将其拖动到新位置。例如，您可以将「目录」放在「图书浏览器」下：

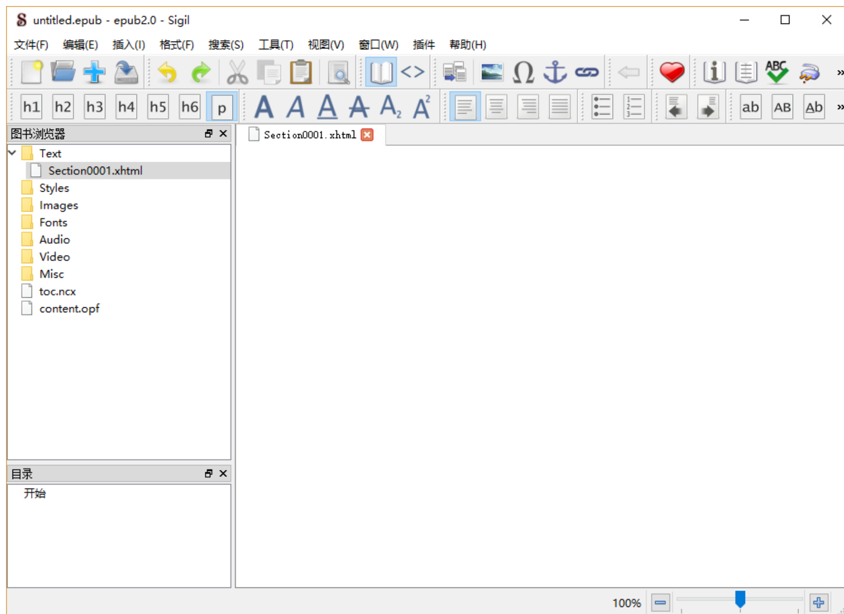


图5-1　Sigil的操作界面

如果将「目录」或者其他窗口直接拖放到「图书浏览器」窗口上，就可以创建选项卡模式，而不是仅在其上方或下方。

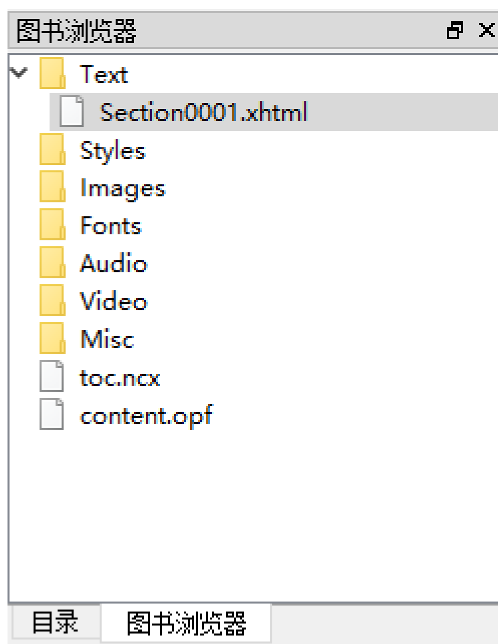


图5-2　「图书浏览器」窗口

你甚至可以分离窗口（将它们与主Sigil窗口分开）。要取消窗口分离，

请双击窗口的标题。在某些操作系统中，您也可以通过点击窗口右上角的小按钮来停用并重新对接窗口。

键盘快捷键

键盘快捷键（例如，用于保存的Ctrl+S）显示在菜单项旁边，并可以在「编辑→首选项：键盘快捷键」中设置。

> 您可以在首选项中创建或更改键盘快捷键。

主菜单

这是始终位于Sigil的窗口或屏幕顶部的菜单项列表：

文件(F) 编辑(E) 插入(I) 格式(F) 搜索(S) 工具(T) 视图(V) 窗口(W) 插件 帮助(H)

图5-3 Sigil的菜单选项

文件

- 新建（Ctrl+N）：创建一个新的、空白的ePub文件。
- 打开（Ctrl+O）：从您的计算机中打开ePub或HTML文件。
- 添加：
  - 现有文件：从您的计算机添加HTML文件、图像等。
  - 空白**HTML**文件：在书中创建一个新的HTML文件。
  - 空白样式表：在书中创建一个新的CSS样式表。
  - 空白**SVG**图片：在本书中创建一个新的SVG图像文件。
- 保存（Ctrl+S）：保存文件（经常使用！）。
- 另存为（Ctrl+Shift+S）：以不同的名称保存文件。
- 保存（Ctrl+S）：保存文件（经常使用！）。
- 保存副本：以不同的名称保存文件副本，而无需重命名当前文件（对备份有用）。
- 打印预览：预览当前选项卡的打印效果。
- 打印（Ctrl+P）：打印当前选项卡。
- 最近打开的文件列表。
- 退出（Ctrl+Q）：退出Sigil，如果尚未保存，询问您是否要先保存文件。

编辑

- 撤消（Ctrl+Z）：恢复刚刚删除、插入或替换的文本。
- 重做（Ctrl+Y）：反转最后一个撤消。

- 剪切：删除所选文本并存储以进行粘贴。
- 复制：复制所选文本并存储以进行粘贴。
- 粘贴：将之前剪切或复制的文本插入到文档中。
- **Edit/Paste From Clipboard History**（Ctrl+Alt+V）：当您使用剪切或复制操作时，文本自动保存在列表中。
- 删除行（Ctrl+D）：删除代码视图中的当前行。
- 改变大小写：
  - 全部小写（Alt+L）：将所选文本更改为小写。
  - 全部大写（Alt+U）：将所选文本更改为大写。
  - 每个单词首字母大写：将每个选定单词的第一个字母更改为大写。
  - 句首字母大写：将选择的第一个字母更改为大写。
- 在光标处拆分（Ctrl+Return）：立即将当前文件分割成两个文件。
- 在章节标记处拆分（F6）：在章节标记处拆分。
- 首选项（F5）：打开「首选项」窗口以更改选项。

插入

- 拆分标记（Ctrl+Shift+Return）：在光标处插入用于分割的标记。
- 文件（Ctrl+Shift+I）：从当前书籍中或电脑中插入图像、音频或视频文件。
- 特殊字符：从特殊字符列表中选择字符并插入。
- **ID**：插入与链接一起使用的锚点名称。
- 链接：在您的文档或其他任何地方中插入链接。
- 关闭标签（Ctrl+.）：在「代码视图」中插入结束标签，以关闭上一个未正确关闭的标签。
- 剪辑：插入一个剪辑并且可以查看该剪辑的快捷键。

格式

- 标题：
  - 标题**1**（Ctrl+1）：将段落格式化为1级标题。
  - 标题**2**（Ctrl+2）：将段落格式化为2级标题。
  - 标题**3**（Ctrl+3）：将段落格式化为3级标题。
  - 标题**4**（Ctrl+4）：将段落格式化为4级标题。
  - 标题**5**（Ctrl+5）：将段落格式化为5级标题。
  - 标题**6**（Ctrl+6）：将段落格式化为6级标题。
  - 普通：将段落格式化为普通文本。
  - 保存现有属性：格式化时保留属性，比如class。
- 粗体（Ctrl+B）：使文字加粗。

- 斜体（Ctrl+I）：使文本斜体。
- 下划线（Ctrl+U）：下划线文字。
- 删除线：给文字中间添加横线。
- 下标：制作文字下标。
- 上标：制作文字上标。

- 居左：将文本与书籍编辑窗口左边对齐（使用删除格式设置可重置为默认值）。
- 居中（Ctrl+E）：将文本在书籍编辑窗口中央对齐。
- 居右：将文本与书籍编辑窗口右边对齐。
- 两端对齐（Ctrl+J）：设置文本内容两端，调整文字的水平间距，使其均匀分布在左右两边之中。

- 项目符号（Ctrl+Shift+L）：将文本更改为前面带有点的列表。
- 编号列表：将文本更改为前面带有数字的列表。

- 减小缩进：减小文本的缩进。
- 增加缩进（Ctrl+Alt+M）：缩进文本。

- 文本方向—左到右：设置文本的方向为从左到右。
- 文本方向—右到左：设置文本的方向为从右到左。
- 文本方向—默认：将文本的方向设置为默认值。

- 显示标签（Ctrl+Alt+T）：在「书籍视图」中显示当前位置的HTML标签。
- 移除格式化（Ctrl+Space）：在「书籍视图」中删除一些样式，同时删除代码视图中的标签。（选中已经格式化的内容，如`<b>`、`<i>`、`<u>`等，然后点击此选项，可以移除格式化标签。）

搜索

- 查找&替换（Ctrl+F）：打开「查找&替换」窗口。

- 查找下一个（Ctrl+G）：向前搜索下一个匹配的文本。
- 查找上一个（Ctrl+Shift+G）：向后搜索下一个匹配的文本。

- 替换（Ctrl+R）：替换任何匹配的文本。
- 替换/查找下一个（Ctrl+]）：替换任何匹配的文本，并向前搜索下一个匹配的文本。
- 替换/查找上一个（Ctrl+[）：替换任何匹配的文本，并向后搜索下一个匹配的文本。
- 替换所有（Alt+A）：根据「查找&替换」设置，替换每个匹配的文本。

- 计数所有：根据「查找&替换」设置，计算匹配的文本数量。

- 当前文件：
    - 文档中查找下一个：无论「查找&替换」模式如何，仅查找当前文件。
    - 文档中替换下一个：无论「查找&替换」模式如何，仅在当前文件中替换/查找下一个。
    - 文档中替换所有：不管「查找&替换」模式如何，仅在当前文件中替换。

    - 文档中计算所有：无论「查找&替换」模式如何，都只在当前文件中计数。

- 书签位置（Ctrl+Alt+B）：在「书籍视图」或「代码视图」里用书签记录您当前的位置。
- 跳转到链接或样式（F3）：跳转到链接对象或跳转到此标签的CSS样式表中的位置。
- 返回（Ctrl+\）：返回到设置书签的位置，单击链接或使用「跳转到链接或样式（F3）」。
- 标识选定文本：在「代码视图」中标记选定的文本，用于「查找&替换」。
- 转到指定行（Ctrl+/）：跳转到「代码视图」中的指定行号。

工具

- 添加封面：选择一个图像作为封面。
- 元数据编辑器（F8）：打开元数据编辑器，编辑书籍详细信息。
- 目录：
    - 生成目录（Ctrl+T）：使用预定义的标题标签（h1~h6）来生成目录。
    - 编辑目录：手动编辑目录。
    - 创建**HTML**目录：创建一个HTML文件版本的目录。
- 索引：
    - 索引编辑器（Ctrl+Alt+I）：管理索引条目。
    - 添加到索引编辑器：将所选文本添加到索引编辑器列表。
    - 索引标记（Ctrl+Shift+X）：在创建索引时将选定的文本添加到索引。
    - 创建索引：创建一个包含索引列表的HTML文件。
- 拼写检查：
    - 拼写检查（Alt+Q）：打开对话框，立即拼写检查所有单词。
    - 高亮拼错的单词：切换到「代码视图」中，突出显示拼写错误的单词。
    - 下一个拼写错误（F4）：在「代码视图」中找到下一个拼写错误的单词。
    - 添加拼错的单词：将拼写错误的单词添加到默认字典中。
    - 忽略拼错的单词：Sigil正在运行时，暂时忽略拼写错误的单词。当您打开新书时，也会清除被忽略的单词。
    - 清除拼错的单词：清除暂时忽略的单词列表。
- **HTML**重新格式化：
    - **Mend and Prettify All HTML Files**：修复所有HTML文件中的HTML语法错误并对其进行格式整理。
    - 改进所有**HTML**文件：修复所有HTML文件中的HTML语法。
- **Epub3 Tools**：
    - **Update Manifest Properties**：更新清单属性。
    - **Generate NCX from Nav**：使用Nav文件中的TOC目录填充至NCX文件（由于不需要内置ncx目录，NCX文件在EPUB3中不会自动保持最新状态）。
- **Well-Formed Check EPUB**：检查您的书是否符合ePub标准。

- 验证样式表：使用W3C Web服务检查您的CSS文件是否符合规范。
- 报告（Ctrl+Shift+R）：显示HTML文件、图像、CSS样式等的细节。
- 剪切板编辑器（Ctrl+Alt+C）：编辑文本和样式的片段并粘贴到文档中。
- 搜索模板（Ctrl+Alt+F）：编辑和运行保存的「查找&替换」项目。
- 删除未使用的媒体文件：从书中删除未链接的图像、视频和音频文件。
- 删除未使用的样式：从CSS文件中删除未使用的样式。

视图

- 工具栏
    - 文件：显示或隐藏「打开和保存」按钮。
    - 编辑：显示或隐藏「撤消和剪切」按钮。
    - 查看：显示或隐藏「书籍和代码视图」按钮。
    - 插入：显示或隐藏「插入章节和图像」按钮。
    - 回退：显示或隐藏「后退」按钮。
    - 捐赠：显示或隐藏「捐赠」按钮。
    - 工具：显示或隐藏「工具」按钮。
    - 标题：显示或隐藏「标题」按钮。
    - 格式：显示或隐藏「粗体和格式化」按钮。
    - 对齐：显示或隐藏「文本对齐」按钮。
    - 列表：显示或隐藏「编号的列表」按钮。
    - 缩进：显示或隐藏「缩进」按钮。
    - 大小写：显示或隐藏「更改大小写」按钮。
    - 文本方向：显示或隐藏「文本方向」按钮（默认情况下关闭）。
    - 剪辑栏：显示或隐藏「剪辑」按钮（默认情况下关闭）。
- 书籍视图：切换到「书籍视图」进行编辑。
- 代码视图：切换到「代码视图」进行编辑。
- 切换视图状态（F2）：切换到「书籍视图」或「代码视图」。
- 放大（Ctrl+＝）：增加视图（书籍视图和代码视图）中文字和图像的大小。
- 缩小（Ctrl+-）：减小视图（书籍视图和代码视图）中文字和图像的大小。
- 缩放重置（Ctrl+0）：将窗口缩放重置为100%。
- 图书浏览器（Alt+F1）：打开或关闭ePub中的文件列表窗口。
- 剪切板：打开或关闭所有剪辑的可选列表窗口。
- 预览（F10）：打开或关闭实时预览窗口。
- 目录（Alt+F3）：打开或关闭可点击的目录窗口。
- 验证结果（Alt+F2）：打开或关闭验证窗口。

窗口

- 下一个标签（Ctrl＋PgUp）：将视图更改为下一个标签。
- 上一个标签（Ctrl＋PgDown）：将视图更改为上一个标签。
- 关闭标签（Ctrl＋W）：关闭当前标签。
- 关闭其他标签（Ctrl＋Alt＋W）：关闭除当前标签之外的所有标签。
- 前一个文件（Alt＋PgUp）：在视图中打开「图书浏览器」中所选文件上方的文件。
- 下一个文件 （Alt＋PgDown）：在视图中打开「图书浏览器」中所选文件下方的文件。

插件

- 插件管理：安装或移除插件。
- 编辑：运行编辑类插件。
- 有效性：运行有效性类插件。
- 输入：运行输入类插件。
- 输出：运行输出类插件。

帮助

- 用户指南（F1）：官方的Sigil用户指南。
- 捐赠：如何捐款以帮助Sigil继续发展。
- **Sigil**网站：Sigil官方网站。
- 关于：Sigil的版本和作者信息。

工具栏

工具栏是窗口顶部的按钮行。但按钮并不是固定在屏幕的顶部——您可以单击并拖动任何按钮组（在这里显示一个未合并的「图书浏览器」和顶部的「目录」窗口）：

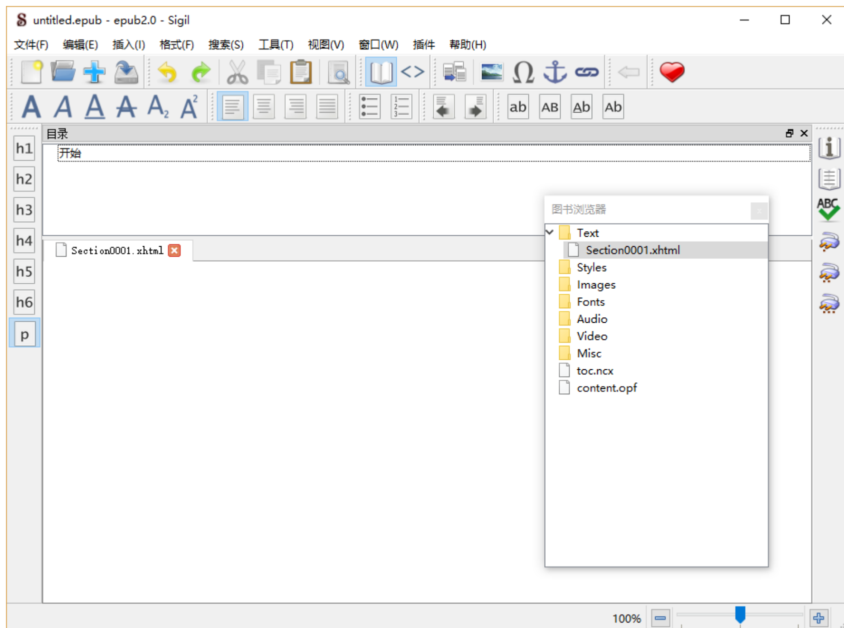


图5-4　工具栏

主工具栏

主工具栏包含Sigil主要功能的快捷键按钮：



图5-5　「工具栏」快捷按钮

> 要移动按钮，请单击并按住每个按钮组左侧的小条，然后将其拖动到您想要的位置。右键单击工具栏，轻松选择要显示的按钮或窗口。

下面列出按钮图标与其等效的菜单项：



- 新建：文件→新建
- 打开：文件→打开
- 添加现有文件：文件→添加→现有文件
- 保存：文件→保存

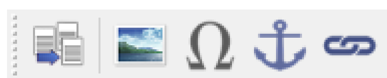


- 撤销：编辑→撤销
- 重做：编辑→重做

- 剪切：编辑→剪切
- 复制：编辑→复制
- 粘贴：编辑→粘贴
- 查找&替换：搜索→查找&替换



- 书籍视图：视图→书籍视图
- 代码视图：视图→代码视图



- 从光标处分割：编辑→在光标处拆分
- 插入文件：插入→文件
- 插入特殊字符：插入→特殊字符
- 插入**ID**：插入→ID
- 插入链接：插入→链接



- 返回：搜索→跳转到链接或样式



- 捐助：帮助→捐赠



- 元数据编辑器：工具→元数据编辑器
- 生成目录：工具→目录→生成目录
- 拼写检查：工具→拼写检查→拼写检查
- **Run Plugin 1**：运行用户分配的插件1
- **Run Plugin 2**：运行用户分配的插件2
- **Run Plugin 2**：运行用户分配的插件3

格式化工具栏

格式化工具栏包含主要用于格式化文本的按钮。这是默认工具栏：



图5-6　「格式化工具栏」快捷按钮

下面列出按钮图标及其等效的菜单项：



- 标题1：格式→标题→标题1
- 标题2：格式→标题→标题2
- 标题3：格式→标题→标题3
- 标题4：格式→标题→标题4
- 标题5：格式→标题→标题5
- 标题6：格式→标题→标题6
- 段落：格式→标题→普通



- 粗体：格式→粗体
- 斜体：格式→斜体
- 下划线：格式→下划线
- 删除线：格式→删除线
- 下标：格式→下标
- 上标：格式→上标



- 左对齐：格式→居左
- 居中：格式→居中
- 右对齐：格式→居右
- 两端对齐：格式→两端对齐



- 符号：格式→项目符号
- 编号：格式→编号列表

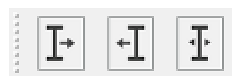


- 减少缩进：格式→减小缩进
- 增加缩进：格式→增加缩进



- 全部小写：格式→改变大小写→小写

- 全部大写：格式→改变大小写→大写
- 每个单词首字母大写：格式→改变大小写→每个单词首字母大写
- 句首字母大写：格式→改变大小写→句首字母大写



- 左至右阅读顺序：格式→文本方向—左到右
- 右至左阅读顺序：格式→文字方向—右到左
- 默认：格式→文本方向—默认

剪辑栏

- 由不在组中的前20个剪辑创建的按钮。

> 默认情况下，「文字方向」和「剪辑栏」不可见。使用「视图→工具栏」，隐藏或显示按钮组。

右键菜单

右键菜单是当您右键单击文本、标签、窗口、选项卡、标题、文本框等时出现的菜单——例如「图书浏览器」中的文件名，「代码视图」中选定的文本或在「图书浏览器」中的图像。右键菜单项会在相关章节中讨论：

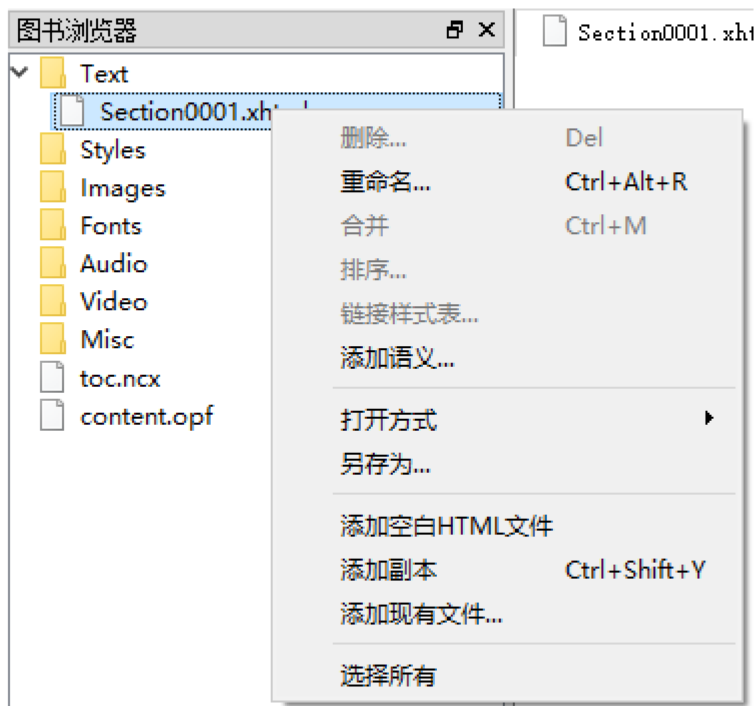


图5-7　「图书浏览器」的右键菜单

> 如有疑问，请右键点击某些内容，看看是否有额外可用的操作！

放大

在主窗口的右下角是一个缩放控件，可让您在「书籍视图」和「代码视图」窗口中独立地缩小放大文本与图像：

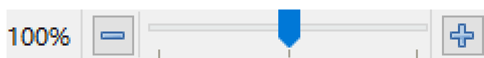


图5-8　缩放控件

100%表示视图的缩放级别，当您单击 + 或 − 图标时，该百分比将增加或减少。

状态栏

在操作完成时，主窗口（位于放大控件左侧）的底部区域可能会列出状态信息。

状态栏还可以显示有关当前打开文件的信息，例如「代码视图」中的行和列：

EPUB已保存。　　　行：10，列：1　100%

图5-9　状态栏

提示

许多按钮、下拉框、标签和字段都有提供有关如何使用它们的信息提示。

要查看工具提示，只需将鼠标悬停在按钮或其他项目上，即可显示帮助文本：

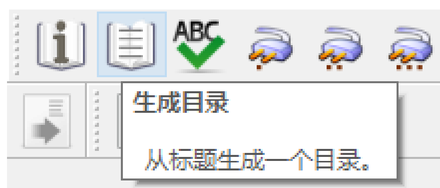


图5-10　悬停提示

> Sigil有许多工具提示——大多数对话框和按钮会提供一些帮助性文字。

5.3.2 首选项：配置Sigil

「首选项」对话框可让您控制Sigil中的许多设置。

选择菜单项「编辑→首选项」，打开「首选项」对话框：

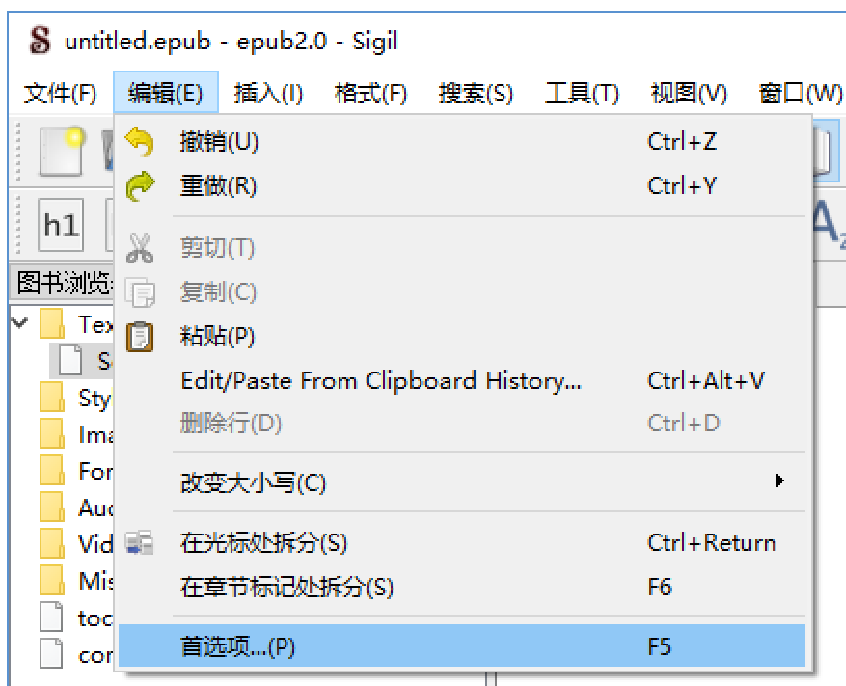


图5-11 「编辑」菜单选项

您可以更改以下设置（有关详细信息，请参阅下文）：

- 外观
- 一般设定
- 键盘快捷键
- 语言
- HTML实体符号保留
- 插件

当「首选项」窗口打开后，您可以通过单击左侧的条目来选择要修改的区域。选择OK或关闭窗口将保存当前设置。

外观

> 外观设置不影响ePub如何在电子阅读器上的显示！

您可以修改Sigil在编辑书籍时显示的字体和颜色：

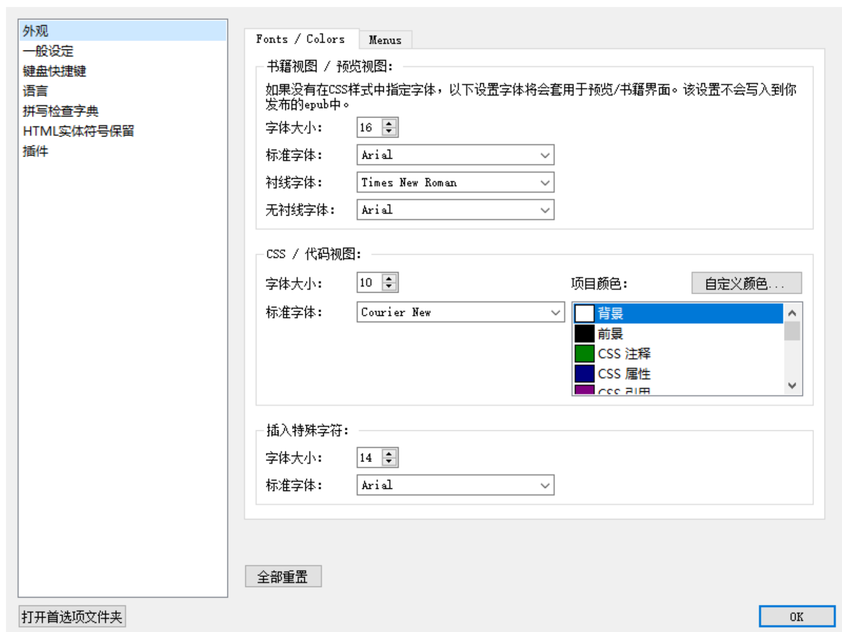


图5-12　「外观」首选项

可以更改的设置有：

- 书籍视图/预览视图：您可以独立设置字体系列/尺寸，标准字体、衬线字体和非衬线字体。
- **CSS**/代码视图：这些设置在代码视图和CSS文件中生效，如字体系列/大小。您还可以设置高亮文本的颜色。
- 插入特殊字符：您可以在「特殊字符」对话框中设置字体系列/大小。

> 「书籍视图」和「代码视图」的默认字体是Arial和Times New Roman。如果您的系统中不存在，「首选项」将默认为第一个可用的字体。

- 主菜单图标大小：此滑块可更改主菜单中图标（工具栏中功能按钮图标）的大小。

键盘快捷键

您可以查看分配给Sigil功能的所有键盘快捷键，并将其改为自己习惯使用的快捷键。所以如果你想要F12总是打开「查找&替换」窗口，那么你可以在这里进行更改。

> 如果您需要在短时间内更容易使用一些功能，也可以暂时分配或重新分配快捷键。

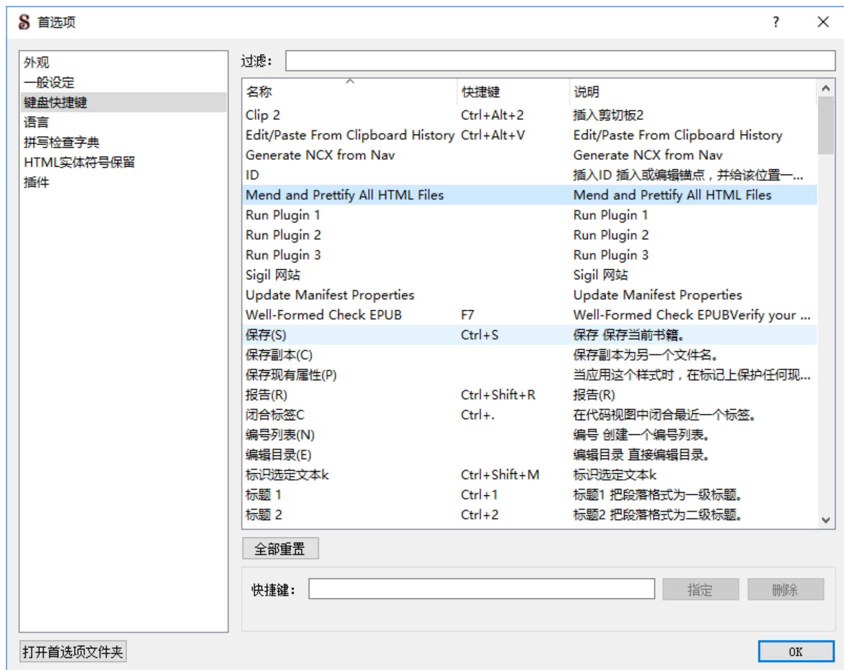


图5-13　「键盘快捷键」首选项

设置或更改快捷键：

1. 从列表中选择条目。
2. 单击快捷键文本框。
3. 键入您要使用的快捷键——不要使用字母拼写，只需按照顺序按键盘。
4. 单击指定设置快捷键（如果与另一个条目有冲突，您将看到一条消息提示）。
5. 您还可以单击删除以删除快捷键——这将删除此条目的任何快捷键。

使用全部重置将所有快捷键改回默认快捷键方式。

> 使用「过滤」文本框轻松搜索快捷键。

语言

您可以设置Sigil的界面语言和书籍的元数据默认语言：

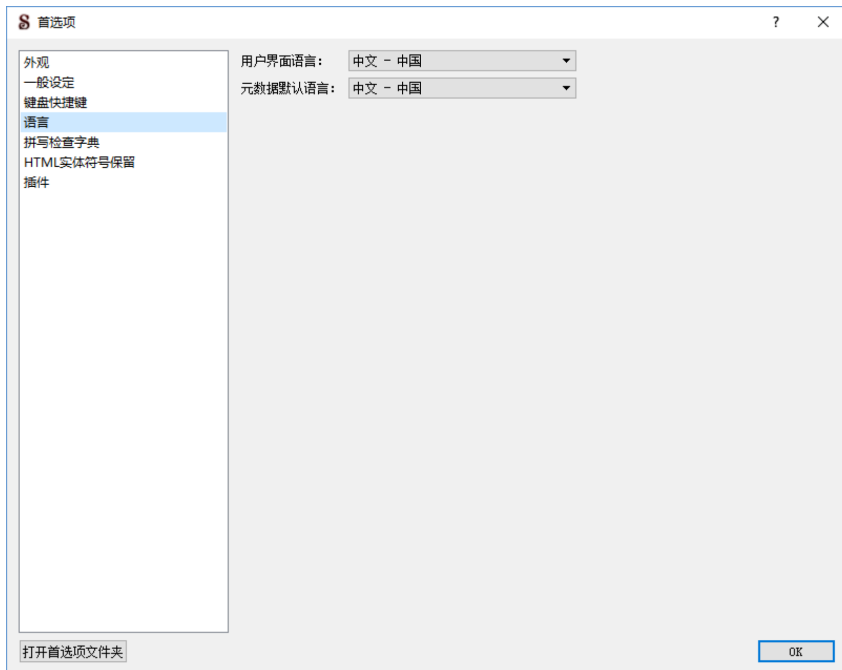


图5-14　「语言」首选项

用户界面语言

Sigil的用户界面语言已经被翻译成多种语言：

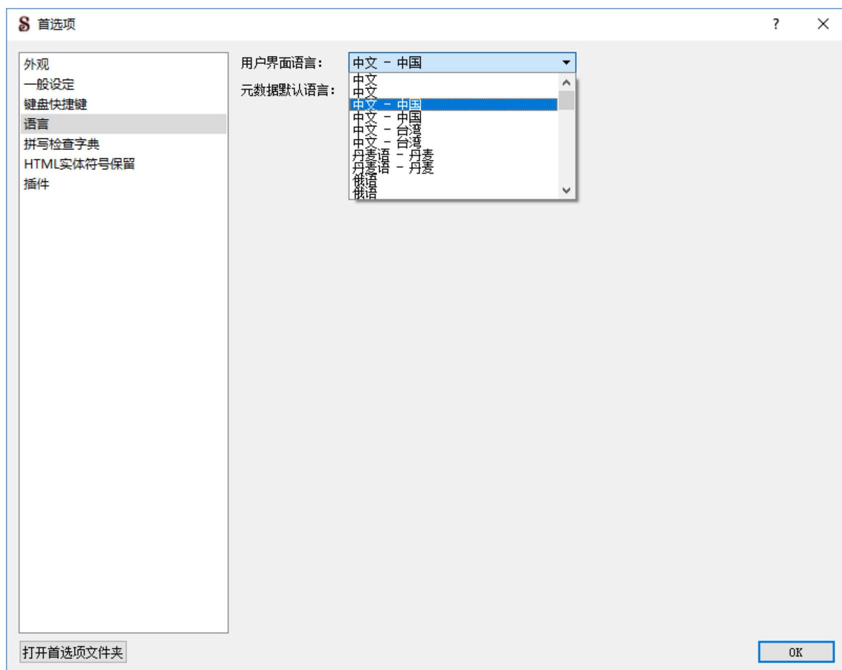


图5-15 「用户界面语言」设置

您可以使用此选项，选择可用语言。如果某些单词不能被标准地翻译，那么它们将以英文显示。

您必须重新启动Sigil才能使设置的用户界面语言生效。
尝试将界面设置为另一种语言，并重新启动Sigil以查看志愿者创建的翻译。

元数据默认语言

此选项在「元数据编辑器」中为尚未设置语言的新书或书籍选择一种语言。

拼写检查词典

Sigil支持「代码视图」中的拼写检查，并允许您选择标准词典和个人用户字典：

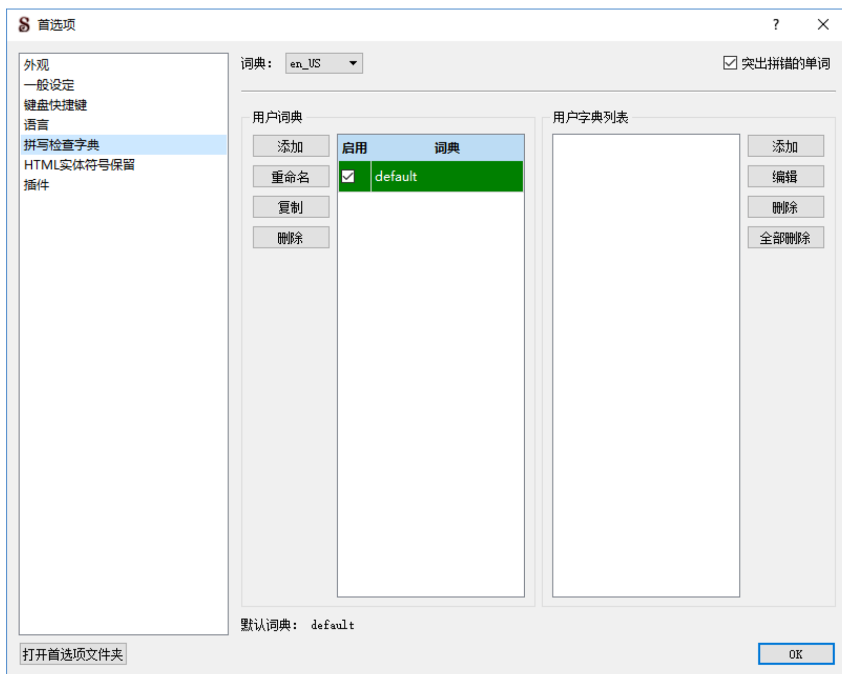


图5-16 「拼写检查字典」首选项

突出拼错的单词

启用此选项，将使用红色波浪下划线突出显示「代码视图」中拼写错误的单词。然后，您可以右键单击该单词，选择建议的单词或将其添加到字典中。

词典

Sigil包括几个标准字典：美国英语、英国英语、法语、德语、西班牙语。您可以使用「词典」下拉菜单选择。

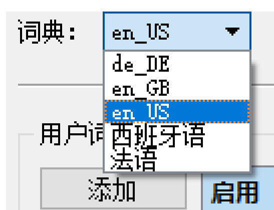


图5-17　选择「词典」

如果要添加新的标准词典（必须使用hunspell格式），则可以单击打开首选项文件夹按钮并进入「hunspell_dictionaries」目录。将您的新词典文件放入「hunspell_dictionaries」目录中。

> 要添加其他标准词典，例如在OpenOffice　Dictionaries站点中找到的词典，请下载，然后解压缩，复制文件，例如：en_GB.aff、en_GB.dic、hyph_en_GB.dic到「hunspell_dictionaries」目录位置，然后重启Sigil。

用户字典

您可以创建自己的个人用户字典，甚至使用不同的书籍和不同类型的单词（地点、名称等）。

如果您尚未创建任何用户字典，Sigil将为您创建并启用「默认」用户字典，您可以进行重命名或删除等操作。

要创建新的用户字典，请在左侧的「用户词典」下选择添加，然后输入新词典的名称：

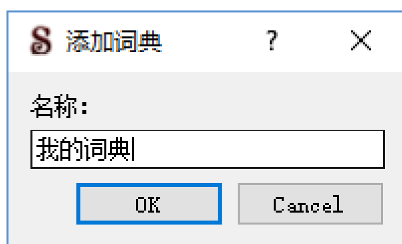


图5-18　添加词典

当您选择左侧的用户字典时，您将看到右侧该字典的单词，您进行添加、编辑或删除等操作。

要直接将字词添加到用户字典中，请选择字典，然后单击右侧的添加。

然后，您可以输入任意数量的文字，或者将它们从文件中粘贴出来——单词可以用逗号、空格或每行列出。

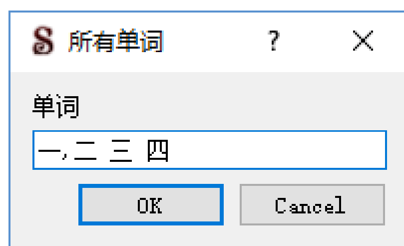


图5-19　添加单词条目

您还可以使用「拼写检查」添加单词到词典中或者右键单击「代码视图」中拼写错误的单词，然后选择「添加到词典」中的任一一个。

> 选择字典使其成为默认字典——当您使用「添加拼错的单词」功能或「添加到默认字典」菜单项添加单词时，将使用该字典。

您可以使用复制按钮，复制整个词典，然后根据需要重新命名。

您可以通过标记词典名称旁边的复选框来选择启用哪些用户字典。检查单词以查看是否拼写错误时，将使用所有启用的字典。

> 您可以一次启用多个字典。

插件

在此窗口中，您可以选择外部Python解释器，安装或卸载插件，或者将插件分配给插件工具栏按钮。

这里有Sigil官网提供的插件列表。

针对「插件」的设置如下：

- **Use bundled Python**：启用此选项以选择捆绑的Python解释器（推荐）。请注意，如果禁用此选项，则需要选择外部Python 3解释器。
- **Interpreter Executable**的路径：此选项适用于插件开发人员；如果禁用Use Bundled Python（或无法选择），您还需要选择它。单击识别或手动指定来选择外部Python 3解释器。
- 添加插件：点击添加插件按钮，安装插件。
- 移除选定：从已安装插件列表中选择一个插件，点击移除选定按钮，卸载插件。
- 全部删除：点击全部删除按钮，卸载已安装插件列表中的所有插件。
- **Assign as Plugin 1**：选择要分配给「Run Plugin 1」工具栏按钮的插件。
- **Assign as Plugin 2**：选择要分配给「Run Plugin 2」工具栏按钮的插件。

- **Assign as Plugin 3**：选择要分配给「Run Plugin 3」工具栏按钮的插件。
- **Assign as Plugin 4**：选择要分配给「Run Plugin 4」工具栏按钮的插件。
- **Assign as Plugin 5**：选择要分配给「Run Plugin 5」工具栏按钮的插件。

打开首选项文件夹

所有首选项条目窗口上的打开首选项文件夹按钮可用于打开包含所有首选项文件的目录。您可以复制文件进行备份或与他人共享。

首选项文件夹可能位于以下位置：

Windows:

```
C:\Users\admin\AppData\Local\sigil-ebook\sigil
```

MAC:

```
~/Library/Application Support/sigil-ebook/sigil/
```

Linux:

```
$HOME/.local/share/sigil-ebook/sigil
```

您可能会看到的文件或文件夹：

- **hunspell_dictionaries**：添加的标准字典存储在这里。
- **local-storage**：本地存储。
- **plugins**：安装的插件位置。
- **plugins_prefs**：配置插件选项的位置。
- **user_dictionaries**：存储个人用户词典的地方。
- **sigil_index.ini**：您的索引条目。
- **sigil_searches.ini**：您保存的搜索。
- **sigil_clips.ini**：你保存的剪辑。
- **sigil.ini**：Sigil的一般配置信息。

5.3.3　打开并保存文件：加载并保存您的书籍

Sigil支持打开ePub文件，也可以阅读HTML和文本文件。当Sigil保存您的书时，它以ePub格式保存：ePub文件将所有HTML文件、图像、样式表、字体等压缩到一个文件中。

> 及时保存，频繁保存，备份副本！
> 在另一个程序将ePub打开之前，请务必关闭Sigil，否则其他程序将锁定文件，并阻止您在Sigil中保存文件。

打开文件

要打开文件，请单击「打开」按钮，或选择菜单项「文件→打开」或其键盘快捷键，然后选择要打开的文件：

ePub文件

- ePub文件中的所有内容都会自动加载：HTML文件、图像、样式表、元数据等。
- Sigil将会整理您的文件到相应的文件夹中，如「图书浏览器」所示。
- 如果ePub包含常见错误，Sigil将提示用户是否选择修复。
- 如果ePub格式化后仍然包含无效的HTML文件，Sigil仍然会加载该文件并报告需要修复的错误。

HTML文件

- 如果可能，HTML文件和任何链接的文件和样式表都将导入，并放入相应的文件夹中。
- HTML文件中的元数据将被加载到该图书的元数据中。
- 如果HTML文件清理后仍然报错，Sigil将报告错误位置，但可能无法加载该文件。您可能需要解决Sigil之外的错误。

文本文件

- 文本文件将被转换为HTML文件。若将它们分割成单独的段落，段落必须用空白行分隔。

添加文件

通过单击「添加现有文件」按钮，选择菜单项「文件→添加→现有文件」或右键单击图书浏览器中的文件，然后选择「添加现有文件」，将文件添加到ePub中：

HTML文件

- 元数据不会从文件加载（与使用「打开」时不同）。
- Sigil将尝试加载引用的文件。

- 文件将被放入相应的文件夹中。

图像文件

- 图像文件将被放置到「Images」文件夹中。

视频文件

- 视频文件将被放入「Video」文件夹。

音频文件

- 音频文件将被放入「Audio」文件夹。

样式表文件

- 样式表文件将被放入「Styles」文件夹中。

字体文件

- 字体文件将被放入「Fonts」文件夹。

其他文件

- 所有其他文件将被放置在「Misc」文件夹中，因为它们通常不会在ePub中使用，尽管可以在Sigil中编辑「Misc」文件夹中的某些文本文件。

添加空白文件

您也可以在Sigil中创建新的空白文件。选择菜单项「文件→添加」，选择空白HTML文件、空白样式表或空白SVG图片（特殊图像格式）。您也可以在「图书浏览器」中添加，甚至可以在首选项中设置键盘快捷键添加。

保存您的ePub

通过单击「保存」按钮保存您的书籍，或选择菜单项「文件→保存」或键盘快捷键。在进行主要修改之前，您应该始终保存文档，以防您要撤消它们。

您保存的ePub文件包含构成图书的所有文件——您的文字、图像、样式表、嵌入式字体、目录、元数据等。

> 如果您编辑了文档中的任何文本，星号（*）则会添加在标题栏的文件名称中。

当您保存书籍时，Sigil会检查它是否存在某些错误，如果需要修复错误，则会发出警告。

> Sigil不会总是能够保存包含错误的ePub，所以频繁的保存将使您更容易识别是否需要更正错误。

保存副本

您可以选择「文件→保存副本」将ePub的副本以不同的文件名进行保存。「保存」或「另存为」之间的区别在于您正在处理的当前书籍未重命名。所以您可以使用此功能来保存您的ePub，而无需更改书籍的名称或保存文件的文件夹。

> 使用「文件→保存副本」进行快速备份。

保存个人文件

您还可以右键单击「图书浏览器」中的文件名，从右键菜单中选择「另存为」。然后，您可以将选定的文件进行命名并保存到计算机上的文件夹中。如果选择多个文件，所有文件将被保存到您选择的文件夹中。这是一种在不共享ePub的情况下与别人共享特定文件的快速方法，也可以创建用于其他图书的模板文件。

5.3.4 书籍视图：所见即所得的编辑

「书籍视图」（或所见即所得模式——您所看到即是您所获得的）是您可以在Sigil编辑书籍的两种方法之一。大多数人在开始使用Sigil编辑他们的书籍时用的就是「书籍视图」，因为它最接近标准文字处理器。当您不太关心基础HTML代码时，它是对您的图书进行基本更改的好方法。但是，当您想要更好地控制ePub的格式或使用Sigil的高级功能时，您可以使用「代码视图」。

> 「书籍视图」只是您的书籍大致预览效果——您需要在阅读器上测试您的书籍。

通过单击「书籍视图」按钮，选择菜单项「视图→书籍视图」或其键盘快捷键切换到「书籍视图」：

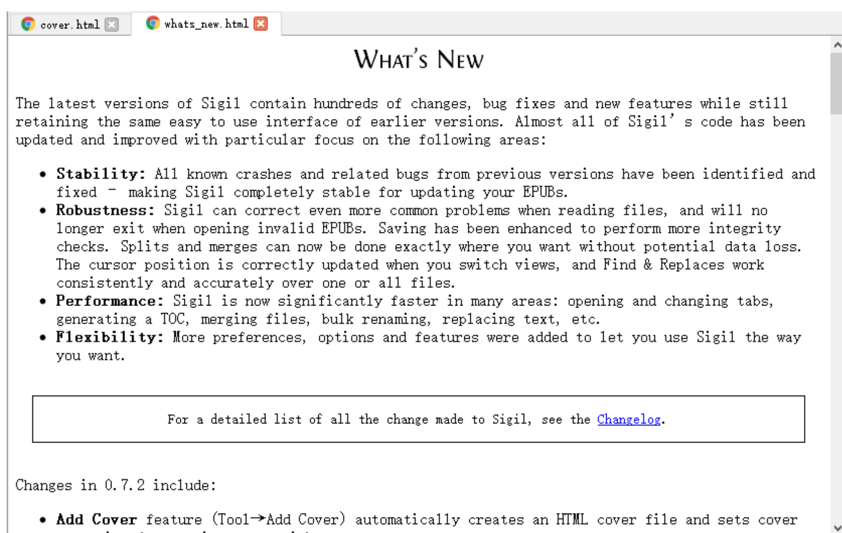


图5-20 「书籍视图」窗口

正如你所看到的那样，文本和图像会显示在屏幕上，就像在电子阅读器上一样。您可以在编辑器中编辑、删除或添加所需的任何文本。

当您在「书籍视图」中应用格式化时，Sigil将尽力应用您想要的格式。然而，在某些情况下，它可能看起来不是你想要的（有的是用HTML标签，有的是用内联式css）。您可以撤消更改，然后尝试重新格式化文本。但是，如果要确保格式化完全符合您的要求，则需要使用「代码视图」。随时可以在「书籍视图」和「代码视图」之间切换，因为它们只是同一页面的两个视图。您当前的段落位置将被记住。

> 您可以使用键盘快捷键F2或双击当前选项卡的名称，在「书籍视图」和「代码视图」之间切换。

Sigil中的某些功能仅在「代码视图」中可用。例如「拼写检查」和「搜索编辑器（查找&替换）」只能在「代码视图」中使用，如果您尝试替换文本或者查找所有HTML文件，「查找&替换」将自动切换到「代码视图」。

格式化

格式化工具栏上的所有按钮都可供您使用。

右键菜单

「书籍视图」中单击右键出现的右键菜单包含一些有用的功能。

右键单击图像可以查看以下选项：

- 打开「选项卡」：在选项卡中打开图像。
- 打开方式：在外部编辑器中打开图像。
- 另存为：将图像保存到计算机中的指定位置。
- 复制图片：复制图片，以便将其粘贴到另一个应用程序中。

右键单击文本有以下选项：

- 剪切板：您定义的任何剪切将在这里显示。
- 添加到剪切板：将所选文本添加到您的剪切板中。
- 插入文件：从书籍或电脑中将图像、视频或音频文件插入到页面中。
- 检查元素：如果未打开「预览」窗口请先打开，以便您可以在「预览」窗口中右键单击选择「检查元素」，查看样式如何应用于文本。

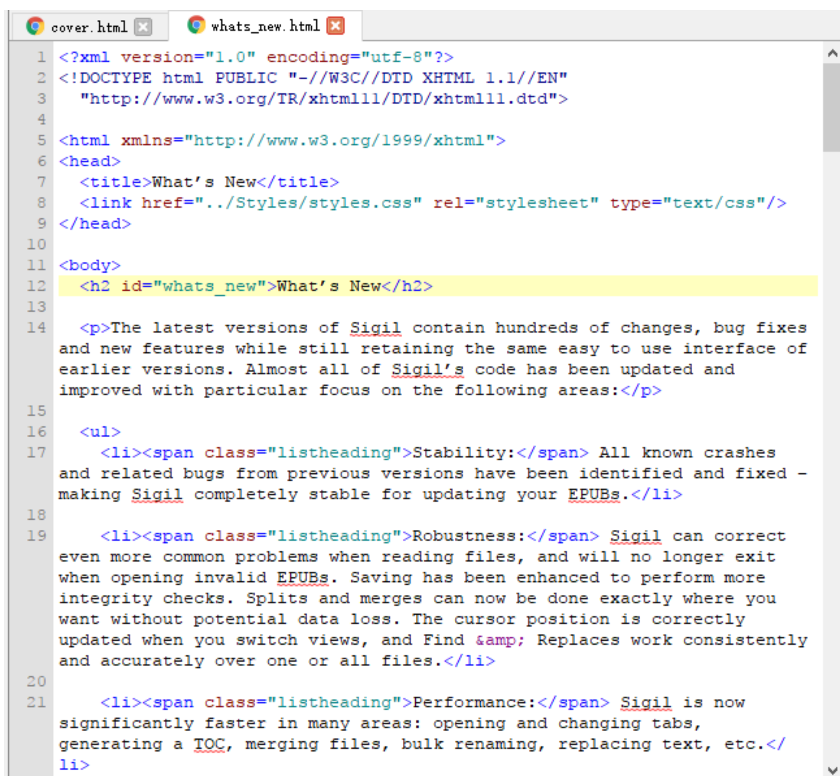
标准文本菜单项目也包括：

- 撤销
- 重做
- 剪切
- 复制
- 粘贴
- 选择所有

5.3.5 代码视图：直接编辑HTML

「代码视图」允许您直接编辑ePub中的HTML代码，这是高级用户编辑文件的最常用方式。

通过单击「代码视图」按钮<>切换到「代码视图」、选择菜单项「视图→代码视图」或其键盘快捷键：



```
cover.html    whats_new.html
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>What's New</title>
  <link href="../Styles/styles.css" rel="stylesheet" type="text/css"/>
</head>

<body>
  <h2 id="whats_new">What's New</h2>

  <p>The latest versions of Sigil contain hundreds of changes, bug fixes
and new features while still retaining the same easy to use interface of
earlier versions. Almost all of Sigil's code has been updated and
improved with particular focus on the following areas:</p>

  <ul>
    <li><span class="listheading">Stability:</span> All known crashes
and related bugs from previous versions have been identified and fixed -
making Sigil completely stable for updating your EPUBs.</li>

    <li><span class="listheading">Robustness:</span> Sigil can correct
even more common problems when reading files, and will no longer exit
when opening invalid EPUBs. Saving has been enhanced to perform more
integrity checks. Splits and merges can now be done exactly where you
want without potential data loss. The cursor position is correctly
updated when you switch views, and Find &amp; Replaces work consistently
and accurately over one or all files.</li>

    <li><span class="listheading">Performance:</span> Sigil is now
significantly faster in many areas: opening and changing tabs,
generating a TOC, merging files, bulk renaming, replacing text, etc.</
li>
```

图5-21 「代码视图」窗口

> 您可以使用键盘快捷键F2或双击当前选项卡的名称，在「书籍视图」和「代码视图」之间切换。
> 您还可以使用「预览」窗口显示当前「代码视图」页面的只读预览版本。

「代码视图」非常强大，但编辑时必须小心，确保输入的是有效的HTML代码。

> 从「代码视图」切换到「书籍视图」或保存文件时，Sigil会检查您的代码是否有重大错误。如果有任何错误，系统将提示您手动或自动修复。通常应选择手动修复问题，以避免自动更正时删除文本。

除了在ePub中完全显示HTML代码之外，为了便于阅读和识别，「代码视图」还会以突出颜色显示标签和元素。

> 您可以使用首选项中更改在代码视图中突出显示的颜色。

格式化

格式化工具栏上的所有按钮都可供您使用。

右键菜单

右键单击文本查看包含以下选项的右键菜单：

- 剪切板：选择您定义的文本剪切。
- 添加到剪切板：将所选文本保存到剪切板中。
- 跳转到链接或样式：与「搜索→跳转到链接或样式」相同。
- 标识选定文本：与「搜索→标记选定文本」相同。也可能显示未标记的文本。
- **HTML**重新格式化：
  - 改进和美化代码：使用您的首选项来清理当前文件。
  - 对所有**HTML**执行纠错并统一代码风格：使用您的首选项来清理所有HTML文件。
  - 纠错：若HTML本身有错误则修复当前文件为有效的HTML。
  - 对所有**HTML**文件执行代码纠错：若HTML本身有错误则修复所有HTML文件为有效的HTML。

右键单击拼写错误的单词有以下选项：

- 建议：从建议的选项中选择要替换的单词。
- 添加到默认字典：将单词保存到首选项中选择的默认字典中。
- 添加到字典：将单词保存到从子菜单中选择的字典中。
- 忽略：跳过此单词的拼写检查，直到Sigil重新启动。

右键单击图像可以查看以下选项：

- 查看图片：在可调整大小的窗口中显示图片。您也可以按住Ctrl单击文件名或使用「跳转到链接或样式」显示图片。查看图片时将调整图片大小以适合窗口，并在编辑时保持打开状态。
- 在新标签页打开图像：在新的标签中打开全分辨率图像。

加上常规的功能选项：

- Undo（撤销）
- Redo（重做）
- Cut（剪切）
- Copy（复制）
- Paste（粘贴）

- Delete（删除）
- Select All（选择所有）

5.3.6 预览：预览代码视图的修改

「预览」是一个可移动的窗口，实时更新「代码视图」编辑页面的只读视图。它与「书籍视图」相同，但可以与「代码视图」同时显示。您还可以在底部的一个小窗口中检查您的代码和样式（或者隐藏窗口）。

您可以使用「预览」显示代码中的更改结果（包括即时反馈的错误），还可以打开CSS样式表以进行「预览」中显示的更改。

通过选择菜单项「视图→预览」或其键盘快捷键激活「预览」窗口：

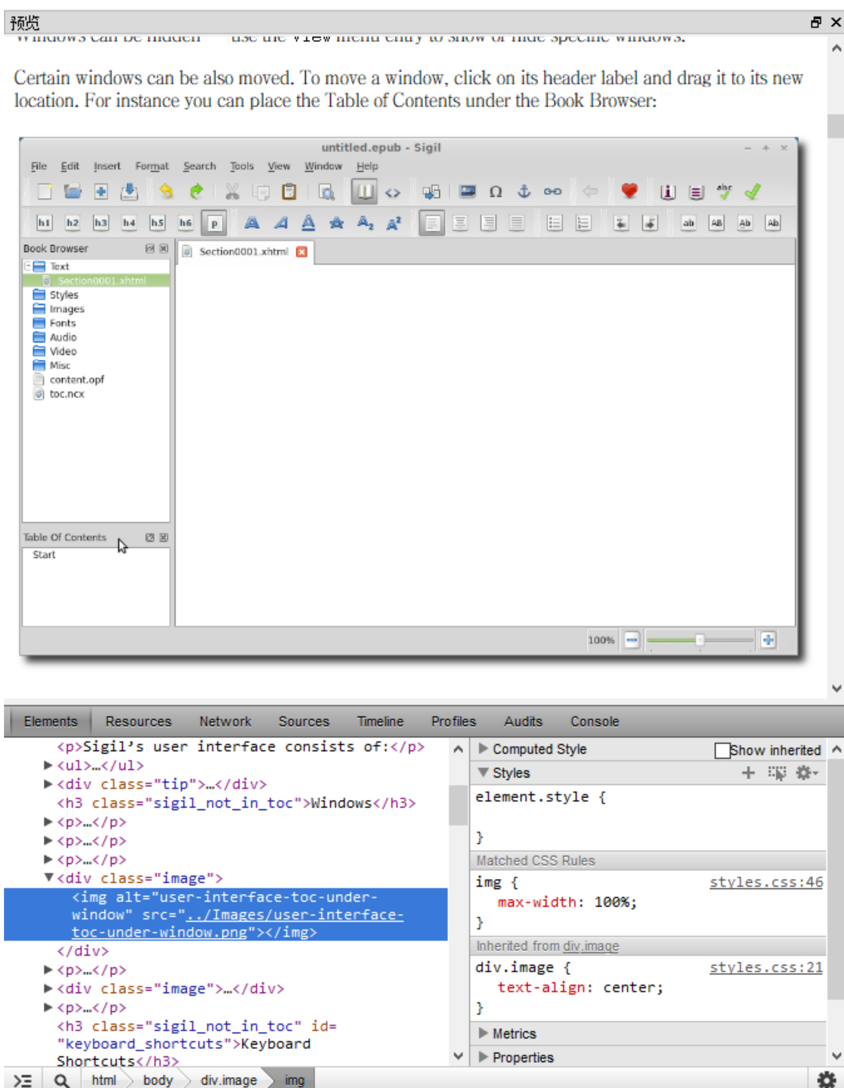


图5-22 「预览」窗口

> 您可以使用其快捷键打开或关闭「预览」窗口，甚至将窗口移动到第二个显示器上（如果有）。

当您单击「代码视图」或「书籍视图」中的某个位置时，「预览」窗口将立即跳转到对应的位置。更新也可以在短时间内停止打字之后。

> 如果您在「预览」窗口中点击某一区域，「代码视图」或「书籍视图」选项卡将同步到同一位置。

右键菜单

「预览」的右键菜单有一个非常有用的功能。

查看此选项需要右键单击屏幕的标题、段落或任何区域：

- 检查元素：在检查器窗口中显示此文本的详细信息。

检查窗口

检查窗口是「预览」页面底部的小窗口。鼠标移动到「预览」窗口底部，鼠标由箭头切换为上下拖动的光标，按住鼠标左键，向上拖动，显示「检查工具」窗口，向下拖动，隐藏「检查工具」窗口。

「检查工具」窗口是一个HTML开发人员的工具，可以详细检查书中的代码。它旨在让您了解特定部分为什么以预定的方式显示，也可以对样式和文本进行临时更改，了解它们如何影响文档。

> 您在「检查工具」窗口中进行的任何更改只是暂时的，不会保存！

「检查工具」基本上是Chrome网页开发人员工具集，您可以从文档中找到有关它的更多信息。并不是所有的信息都适用于Sigil，因为软件是用于一般的网页检查。向下滚动到「开发人员窗口」部分，查看可用的功能：

- Inspector for Chrome Developer Tools

使用「检查工具」的最简单方法是右键单击顶部窗口中感兴趣的文本，然后选择「检查元素」。

使用「检查工具」的另一种方法是单击「检查工具」底部左下角的放大镜按钮，然后将预览区域滚动到要检查的位置并单击它。「检查工具」窗格将滚动到相关的代码段，允许您查看有关文本的各种详细信息。

然后，您可以对文本使用的样式进行临时修改以进行测试。样式的更改即时反映在「预览」区域中，但如果您编辑任何文本，则可能需要在「预览」区域中单击以刷新它。然后，您可以更新您的CSS样式表，以包括您决定保留的任何更改。

> 如果您单击「检查工具」部分中的某个区域，预览区域将不会像您预期的那样进行同步。

CSS样式表预览

您可以看到CSS样式表中的更改如何在「预览」中影响文本。

要使用此功能，请打开HTML页面并打开预览。然后打开你的CSS样式表。

当您在样式表中编辑样式时，在您停止输入一秒钟后，更改将反映在「预览」中的HTML文件中。

5.3.7　图书浏览器：管理ePub中的文件

「图书浏览器」可以访问组成ePub的所有文件。您的文本文件、图像以及任何字体或样式表或其他文件都被组织到文件夹中——允许双击打开和编辑它们或者右键单击它们来选择要执行的操作。

「图书浏览器」通常是打开的，但您可以通过选择菜单项「视图→图书浏览器」关闭它：

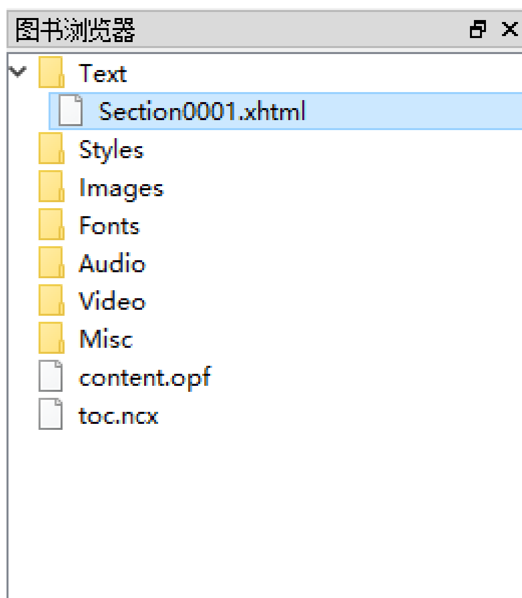


图5-23　「图书浏览器」窗口

> 双击文件将在编辑窗口的选项卡中打开它。
> 右键单击文件名将显示选项的右键菜单。

文件夹和文件

「图书浏览器」将您的文件组织成多个文件夹，可以通过单击文件夹名称左侧的三角来展开或折叠「图书浏览器」，或双击文件夹名称。文件夹和文件包括：

- **Text**：组成图书文本的XHTML、HTML、XML、HTM文件。
- **Styles**：控制书籍样式的CSS样式表。
- **Images**：本书中的PNG、JPG、GIF、SVG图片文件。
- **Fonts**：嵌入在书中的TTF和OTF字体。
- **Audio**：嵌入在书中的MP3、AAC、MPG、MPEG、M4A音频文件。

- **Video**：嵌入在书中的MP4、OGG、WEBM、M4V视频文件。
- **Misc**：其他文件，ePub阅读器通常不会使用它们，尽管它们可以与书一起存储。
- **toc.ncx**：包含本书目录的文件。
- **content.opf**：包含书中文件列表的文件。

要打开文件，只需双击文件名。如果文件尚未在选项卡中打开，则将为其创建一个新选项卡。如果已经在选项卡中打开，视图将切换到该文件的选项卡。

> 如果您更改选项卡，在「图书浏览器」中将切换为该选项卡使用的文件。
> 每个文件夹的工具提示都会显示文件夹中的文件数目。每个文件的工具提示显示完整的文件名和任何指导语义类型。

选择多个文件

您可以在「图书浏览器」中选择多个文件，通过单击一个条目，然后按住Shift键同时单击另一个条目以选择两个条目之间的所有条目。按住Ctrl键同时单击多个条目，将选择多个文件。您还可以单击右键，使用右键菜单中的「全选」轻松选择所有条目。选择文件后，您可以单击右键，使用右键菜单对所有选定文件执行操作。

您只能选择同一文件夹中的多个项目。

重新排序文件

Text文件夹中文件的顺序很重要，因为它是向读者显示文件的顺序。

如果您需要重新排列文件在书中出现的顺序，拖动Text文件夹中的文件，将其上下移动。

> 您还可以使用右键菜单来重命名或重新排序文件。

所有其他文件夹中的文件按字母顺序自动排序，无法重新排列——其文件的顺序不会影响书籍的显示方式。

右键菜单

右键单击「图书浏览器」中的一个或多个项目，会弹出一个包含相关操作的右键菜单：

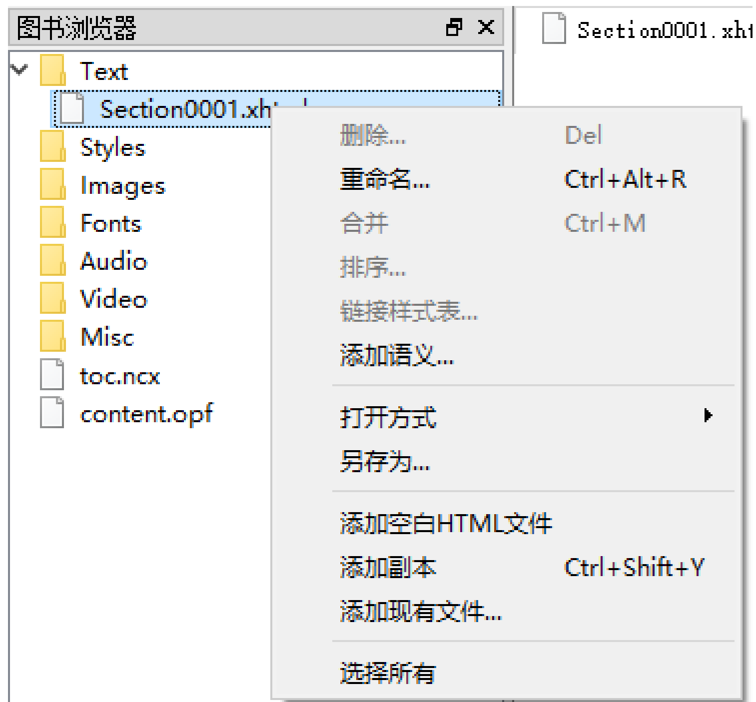


图5-24　内容文档右键菜单

以下部分描述可用的右键菜单功能。

删除

删除所选文件。您不能删除最后剩下的那个HTML文件。

系统将提示您确认要删除文件，因为该操作无法撤消。

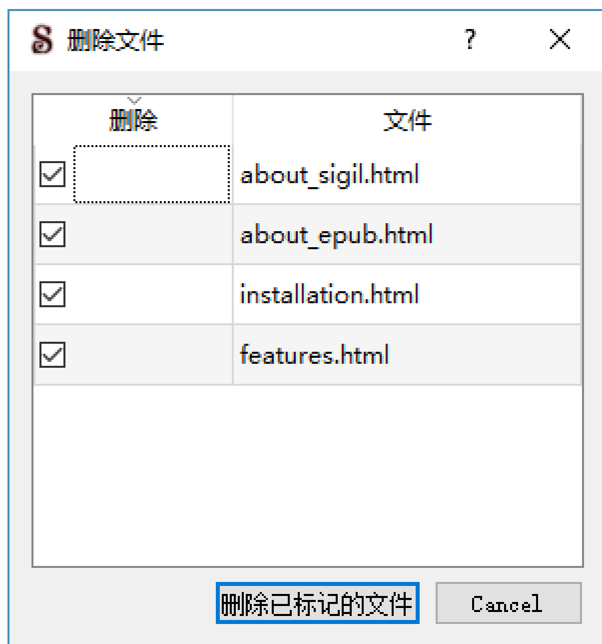


图5-25　删除内容文档

如果您的文件包含在目录（NCX）中，请记住更新或重新「生成目录」。

重命名

如果您选择要重命名的单个文件，则会突出显示文件的名称，以便输入其新名称。默认情况下，扩展名不会突出显示，这使得重命名更容易，但如果需要，还可以更改扩展名。

如果选择多个文件，则会显示一个新对话框，询问您是否要重命名文件：

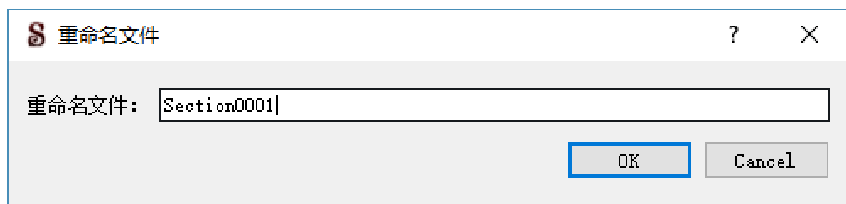


图5-26　重命名内容文档

输入第一个文件的名称，其中包括名称末尾的数字。然后，Sigil将使用您提供的号码开始顺序地命名每个文件。如果重命名文件导致书籍具有重复的文件名，文件将不能被重命名。您也可以重命名文件扩展名。

例如：

表10　Sigil中内容文档命名规则

输入文本	文件重命名	备注
Chapter 1	Chapter 1 Chapter 2 Chapter 3 ……	不会更改文件扩展名
Section_0099	Section_0099 Section 0100 Section_0101 ……	
Part 2–Section 08	Part 2–Section 08 Part 2–Section 09 Part 2–Section 10 ……	
.xhtml	.xhtml	对所有文件的文件扩展名进行重命名
Section_0099.xhtml	Section_0099.xhtml Section_0100.xhtml Section_0101.xhtml ……	

> 当您重命名文件时，指向ePub中文件的任何链接都将被更新。

合并

如果选择单个HTML文件，则该文件将被合并到文件列表中的上一个HTML文件中。

如果您选择多个HTML文件，则所有文件将被合并到您选择的第一个HTML文件中。

您也可以使用合并键盘快捷键。有关详细信息，请参阅拆分与合并一章。

排序

根据数字方式对选定的HTML文件进行排序。这意味着包含数字的文件名将按其预定的数字顺序进行排序，例如。「1 3 2 10」将被排序为「1 2 3 10」而不是默认的「1 10 2 3」。

链接样式表

将样式表链接到选定的HTML文件。选择要从对话窗口链接的样式表——只有标有勾号的样式表才会被链接，其他样式表将与HTML文件取消链接。

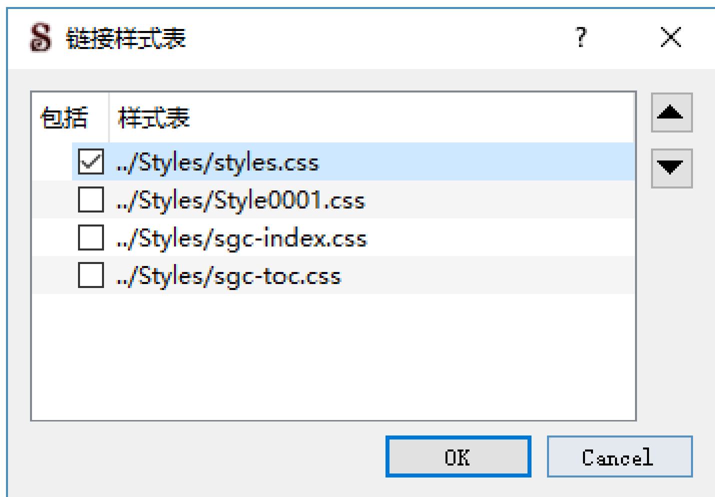


图5-27　链接样式表

> 当对话框首次打开时，只有已经链接样式表的文件才会显示一个复选标记。
> 样式表以对话框中列出的顺序应用到文档中。

添加语义

此子菜单允许您为所选文件指定语义「类型」。ePub电子阅读器可以使用这些信息（存储在`content.opf`文件中）来向读者如何呈现这些文件。

对于图像文件，唯一的选项是封面——表示该图像是本书的封面图片。

对于HTML文件，您可以从以下选项中进行选择——每个文件只能有一种类型：

- Cover 封面
- TitlePage 扉页
- Table Of Contents 目录
- Index 索引
- Glossary 词汇
- Acknowledgements 致谢
- Bibliography 参考书目
- Colophon 后记
- Copyright Page 版权页
- Dedication 贡献
- Epigraph 题词
- Foreword 前言
- List Of Illustrations 插图列表
- List Of Tables 表格列表

- Notes 注释
- Preface 序
- Text 文本

> 应该使用「文本」语义标记来表示当书第一次打开时应该显示的页面，但并不是所有的阅读器都遵循这个约定。

打开方式

打开一个对话框窗口，允许您选择外部应用程序来编辑此文件。

有关详细信息，请参阅外部编辑器一章。

另存为

允许您将所选文件保存到计算机中，便于您需要在ePub之外查看它们。

添加空白HTML文件/样式表/SVG图像

在文件夹中添加一个空白的HTML文件、CSS样式表或SVG图像文件。对于HTML文件，如果您是右键单击文件夹名称添加的，它将添加在文件夹的底部，或者在您右键单击的文件下方。对于其他文件夹，它将按字母顺序添加它们。

这与主菜单项「文件→添加」功能相同。

新文件将从最低可用编号0001开始（例如*Section0001.xhtml*）进行编号，当然也可以进行重命名。

添加副本

创建您点击的HTML文件的副本，并将其添加到该文件的下方。

添加现有文件

打开一个对话窗口，允许您从计算机中选择一个或多个文件以添加到ePub中。

无论您选择哪种类型的文件，文件将根据文件扩展名添加到相应的文件夹中。

这与主菜单项「文件→添加→现有文件」和「添加现有文件」工具栏按钮相同。

选择所有

快速选择当前文件夹中的所有文件，以便您可以一次重命名、链接样式表等。

字体模糊

如果您在电子书中使用自定义字体，则Fonts文件夹的子菜单允许您定义嵌入字体如何存储在ePub中。通常当将字体添加到ePub时，字体以可读的形式存储。在某些情况下，您可能不想让字体可供读者直接访问。在这些情况下，您可以模糊或扰乱电子书中的字体，以便读者无法访问字体，但是ePub阅读器仍然可以使用这些字体。模糊及其含义的细节超出了本指南的范围，但在网络上有很多相关内容。

选项：

- 无
- 使用Adobe的方法
- 使用IDPF的方法

如果所有选定的字体都使用该方法，该方法将在其旁边有一个选择标记。

重新编排目录项

一个高级功能，从1开始顺序的重新创建toc.ncx文件中「内容表」条目的playOrder和navPoint　id序号。如果必须手动添加或删除一个条目，则清除目录编号。

您还可以使用编辑目录功能——只需保存目录，这将重新编号id。

文件中的任何navList或pageList项目都不会被保存——仅在重新编号后保存目录条目。

5.3.8　元数据：编辑书籍的详细信息

「元数据编辑器」允许您添加或编辑图书的详细信息，如作者、标题、语言以及由ePub标准定义的数百个其他条目。

通过单击「元数据编辑器」按钮、选择菜单项「工具→元数据编辑器」或其键盘快捷键来编辑图书详细信息：

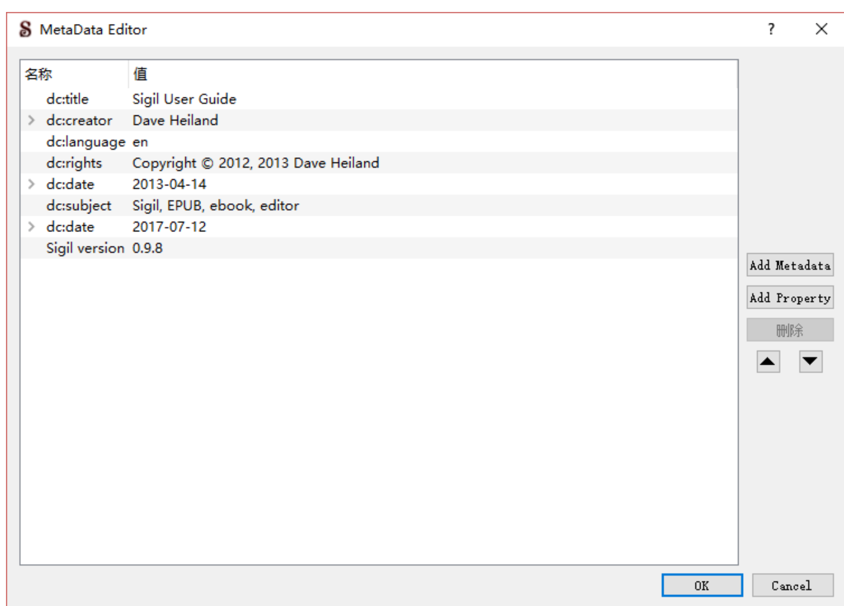


图5-28　元数据编辑器

关于这本书的所有细节（称为元数据）都存储在ePub中的`contents.opf`中。「元数据编辑器」允许您更容易地编辑此信息。

所需详细信息

「元数据编辑器」显示了ePub所必需的3个字段，您必须填写这些字段才能创建有效的ePub：

- 标题：图书的标题（一般为书名）。
- 作者：本书的作者。
- 语言：该书使用的语言。

您可以使用File-As字段列出作者的名称以进行排序，通常为「姓氏　名字」。

可选详细信息

您可以使用Add Metadata或Add Property添加可选的详细信息。这些将被添加到表中。该表允许您为每个条目提供以下值：

- **Name**：字段的名称。
- **Value**：字段的内容。
- **File As**：某些字段，如作者、日期等替代值，用于排序等。
- **Role Type**：所有角色，如作者，可以是创作者或贡献者。

「Add Metadata」包括大约20个项目，如：

- **Date**：日期
- **Date:Creation**：创作日期
- **Date (custom)**：您可以在「File As」列中输入自定义的日期类型。
- **Identifier**：ISBN号码。
- **Identifier (custom)**：您可以在「File As」列中输入自定义的标识符类型。
- **Rights**：版权信息，如Copyright 2012 作者。

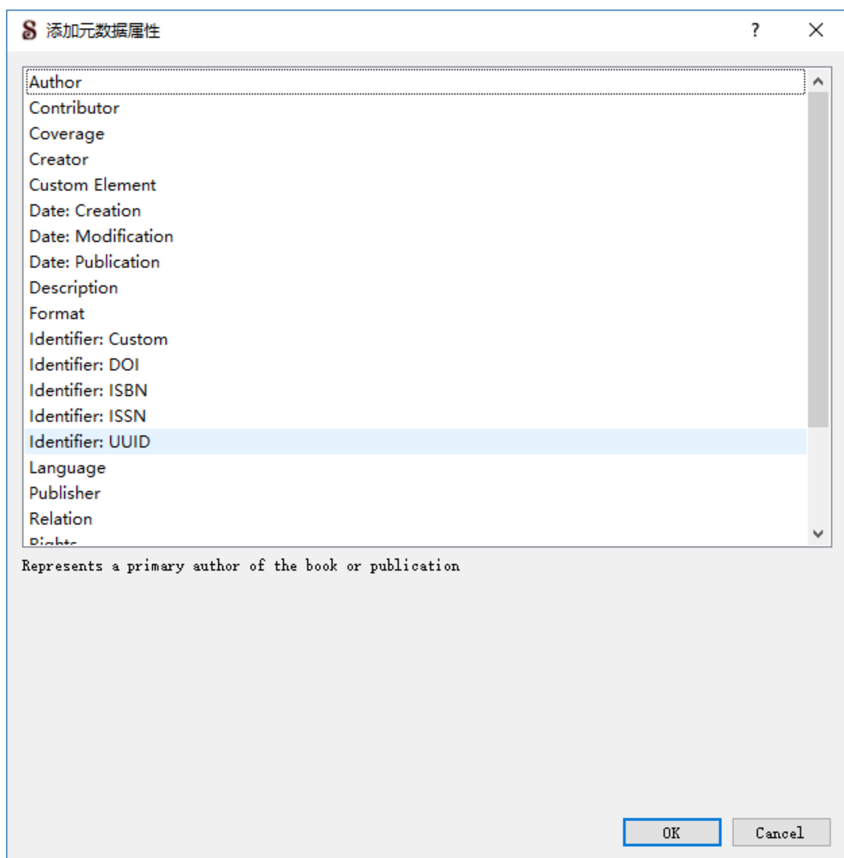


图5-29　添加Metadata元数据属性

Metadata条目：

- Author 作者
- Contributor 合作者/贡献者
- Coverage 覆盖/所包括的范围
- Creator 创造者
- Custom Element 自定义元素
- Date:Creation 创建日期
- Date:Modification 修改日期
- Date:Publication 出版日期
- Description 内容描述
- Format 格式
- Identifier:Custom 自定义标识符
- Identifier:DOI 数字对象标识符
- Identifier:ISBN 国际标准书号
- Identifier:ISSN 国际标准连续出版物编号
- Identifier:UUID Sigil创建的通用唯一标识符
- Language 出版物语言
- Publisher 出版社
- Relation 关联
- Rights 声明
- Source 来源
- Subject 主题词
- Title 标题
- Type 科目

代表本书或出版物的主要作者。

Property条目：

- Custom Attribute 自定义属性
- Event 事件
- File as 归档
- Id Attribute 自定义ID
- Role 角色
- Scheme 方案
- XML Language XML语言

可以修改的元数据属性。

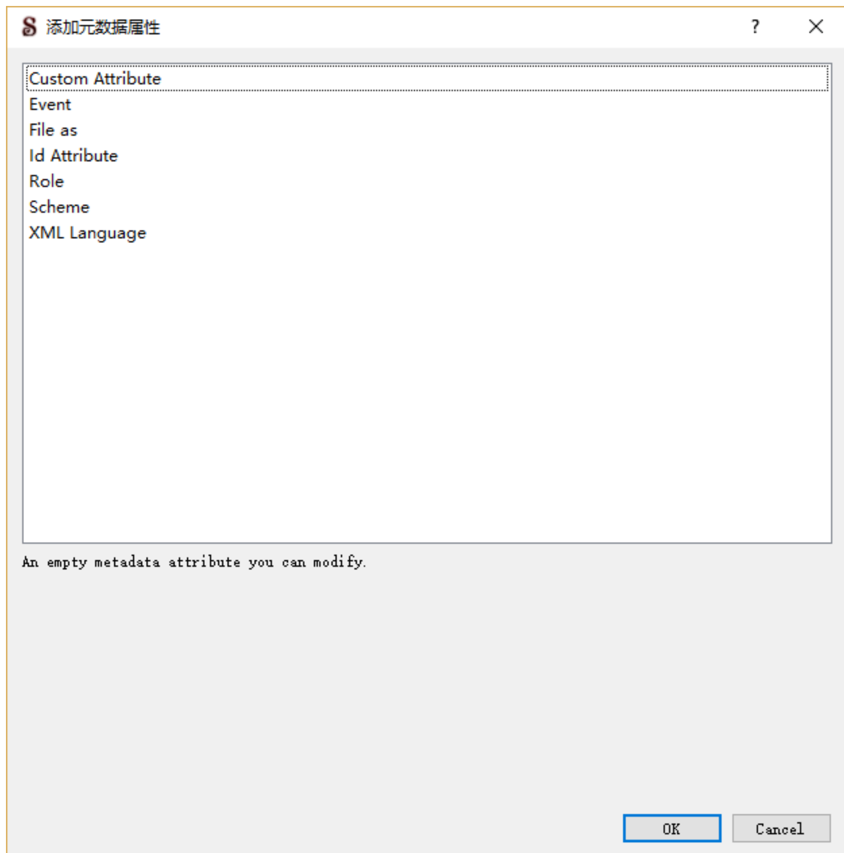


图5-30　添加Property元数据属性

「Add　Property→Role」包含人们在创建本书时可能做的数百种操作，例如作者、编辑者、校对者等。

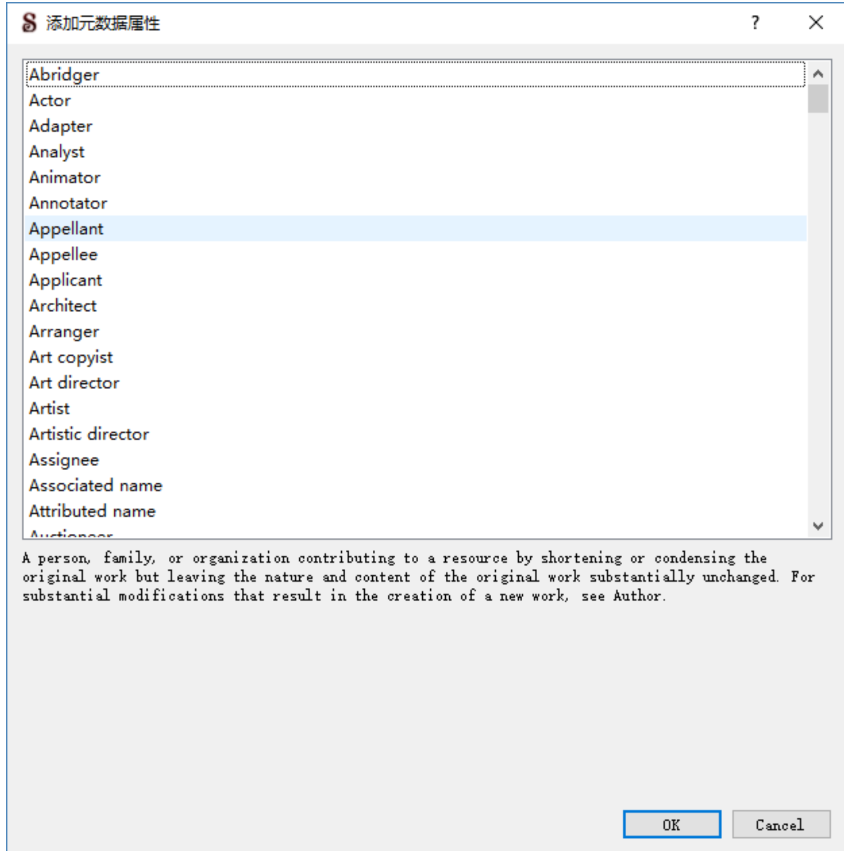


图5-31　Property条目

您可以多次添加相同类型的字段。

> 您可以选择条目并单击向上或向下按钮来重新排列条目的顺序（可能影响ereaders显示信息的方式）。

5.3.9 添加封面：为您的书选一张封面

「添加封面」功能允许您选择封面的图像，并让Sigil自动将图像标记为封面。

通过选择菜单栏「工具→添加封面」或其键盘快捷键添加封面：

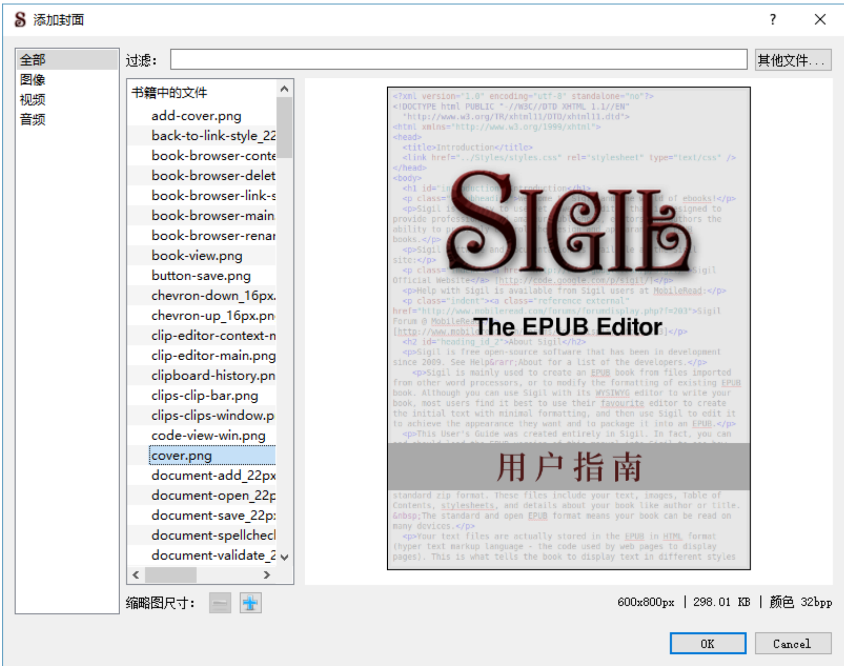


图5-32　添加封面

您可以在epub中选择一个图片，或使用其他文件从计算机中选择一个。

细节

当您添加封面时，Sigil进行以下更改：

- 创建一个「cover.xhtml」文件（如果不存在）。如果它存在，它将使用该文件。如果另一个具有「封面」语义的文件存在，它将使用该文件。
- 插入必要的HTML代码，显示和调整您的图片到HTML文件中。
- 标记HTML文件的语义为「封面」文件。
- 标记图像文件的语义为「封面图片」。

自定义HTML封面文件

Sigil在HTML封面文件中插入代码，允许您的封面在大多数电子阅读器上重新调整大小并显示，但您可能希望使用不同的代码。您可以在创建

文件后编辑代码或用Sigil使用自己的代码。

这是Sigil默认插入到封面文件中的HTML代码：

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops">
<head>
  <title>Cover</title>
</head>
<body>
  <div style="text-align: center; padding: 0pt; margin: 0pt;">
    <svg xmlns="http://www.w3.org/2000/svg" height="100%" preserveAspectRatio="xMidYMid meet" version="1.1" viewBox="0 0 1536 2048" width="100%" xmlns:xlink="http://www.w3.org/1999/xlink"><image width="1536" height="2048" xlink:href="../Images/cover.jpg"/></svg>
  </div>
</body>
</html>
```

要修改Sigil用于HTML封面文件的模板代码，请在您的「首选项文件夹」中创建一个名为「cover.xhtml」的文件

您可以在文本中使用某些占位符，Sigil将用适当的值替换它们：

- **SGC_IMAGE_FILENAME**：这将被ePub中图像的路径替代。
- **SGC_IMAGE_WIDTH**：这将被图像的宽度（以像素为单位）所代替。
- **SGC_IMAGE_HEIGHT**：这将被图像的高度（以像素为单位）所代替。

例如：

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops">
<head>
  <title>Cover</title>
</head>
<body>
  <div><img alt="封面" width="SGC_IMAGE_WIDTH" height="SGC_IMAGE_HEIGHT" src="SGC_IMAGE
></div>
</body>
</html>
```

5.3.10 目录：一本书的地图

ePub支持「目录」（简写为TOC），帮助读者阅读书籍。通常ePub电子阅读器会提供一个按钮或菜单项来打开「目录」。

Sigil提供了几种使用目录的方法。Sigil中的「目录」窗口为您提供了目录的动态视图——您可以准确了解目录的外观，并选择要跳转的章节。「生成目录」功能可以根据文档中的标题标签创建一个新的目录。当您不能或不想使用「生成目录」时，「编辑目录」功能可以更新现有的目录。「创建HTML目录」可以在您的文档中创建一个新的HTML页面，显示ePub目录中的内容。

目录窗口

如果尚未打开「目录」窗口，请选择菜单项目「视图→目录」：



图5-33 「目录」窗口

您可以单击一个条目以跳转到在本书中的位置，如果有子章，则可以通过单击章名称左侧的图标来展开或折叠条目。

> 选择条目，单击右键，选择「全部折叠」或「全部展开」，快速显示或隐藏子章。

ePub目录存储在「图书浏览器」中的*toc.ncx*文件中。

生成目录

为了可以自动创建目录，您首先需要定义书中的标题或章节。

在「书籍视图」中，您可以通过选择文本来创建标题，然后从「格式→标题」菜单中选择所需标题的级别，或者更轻松地在「书籍视图」或「代码视图」中使用工具栏上的按钮：

h1 h2 h3 h4 h5 h6 p

图5-34　「标题格式化」快捷按钮

> 如果您的章节名称不像上述那样使用标题标签（h1，h2等），那么您将无法自动生成目录。

通过单击「生成目录」按钮启动生成目录，或选择菜单项「工具→目录→生成目录」，然后选择要包含在本书目录中的标题：

生成目录

章节	层级	包括
Cover	h1	☑
Title Page	h1	☑
Copyright	h1	☑
介绍 Introduction	h1	☑
更新 What's New	h2	☑
关于Sigil	h2	☑
关于EPUB	h2	☑
安装 Installation	h1	☑
Windows	h3	☑
Mac	h3	☑
Linux	h3	☑
功能 Features	h1	☑
编辑 Editing	h3	☑
查找和替换 Find & Replace	h3	☑
管理文件 Managing Files	h3	☑
目录 Table of Contents	h3	☑
修改图书详细信息 Modifying Book Details	h3	☑

重命名

☑ 仅显示目录项

<选择要包括的标题>

OK　Cancel

图5-35　生成目录

检查您希望包含在目录中的任何标题，并取消选中您不希望包含的任何条目。

> 当您使用「生成目录」时，Sigil不仅会更新您的目录还会更新您的HTML文档，以确保在下次生成目录时可以再次使用您输入的信息。

您可以使用仅显示目录项复选框来显示或隐藏不包含在目录中的条目。这不会影响您的目录，只会在对话框中看到。您还可以使用选择要包括的标题下拉菜单，快速选择或取消选择某个层级的标题，然后手动微调选择。

您可以通过单击目录条目，重命名目录。这也将改变您的HTML代码，以设置标题的「title」属性。

您可以使用左右箭头键更改标题的级别。这也将改变HTML代码中的标题级别。

> 每个标题的工具提示（只需将鼠标移到标题上并停留几秒钟）就可以显示标题所在的文件及其HTML代码。

目录标题的工具提示将显示每个类型的标题数：

层级	包括	隐藏
h1	9	0
h2	50	0
h3	187	0
h4	53	0
h5	0	0
h6	0	0

图5-36　标题数目提示

当您选择OK时，如果它们没有常规ID，Sigil将会自动添加ID或更新您选择的标题ID。然后，它将创建一个链接到新的「toc.ncx」文件中。如果您的标题在HTML文件的开始处，Sigil将直接链接到文件，而不是标题ID。这是因为在一些直接链接到元素的电子阅读器上，会降低目录的显示。

> 自动生成的目录将覆盖「toc.ncx」文件中的任何现有目录信息。

编辑目录

如果您不能或不想用文档中的标题生成目录，则可以使用「编辑目录」来更改现有的目录。对目录所做的任何更改都不会影响您的HTML文件，如果使用「生成目录」，则所有更改都将丢失。

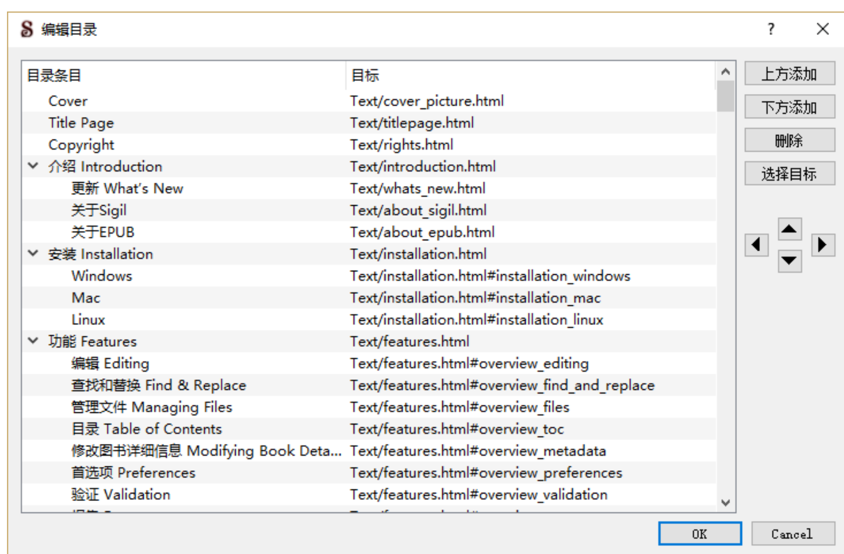


图5-37　编辑目录

双击它可以编辑一个条目。您可以添加新条目或删除现有条目。使用左右箭头增大或减小条目的标题级别。

要更改目录条目指向的内容，可以直接键入目标目标，或使用选择目标按钮。这将打开「选择目标」对话框，您可以在其中选择目录条目指向的位置：

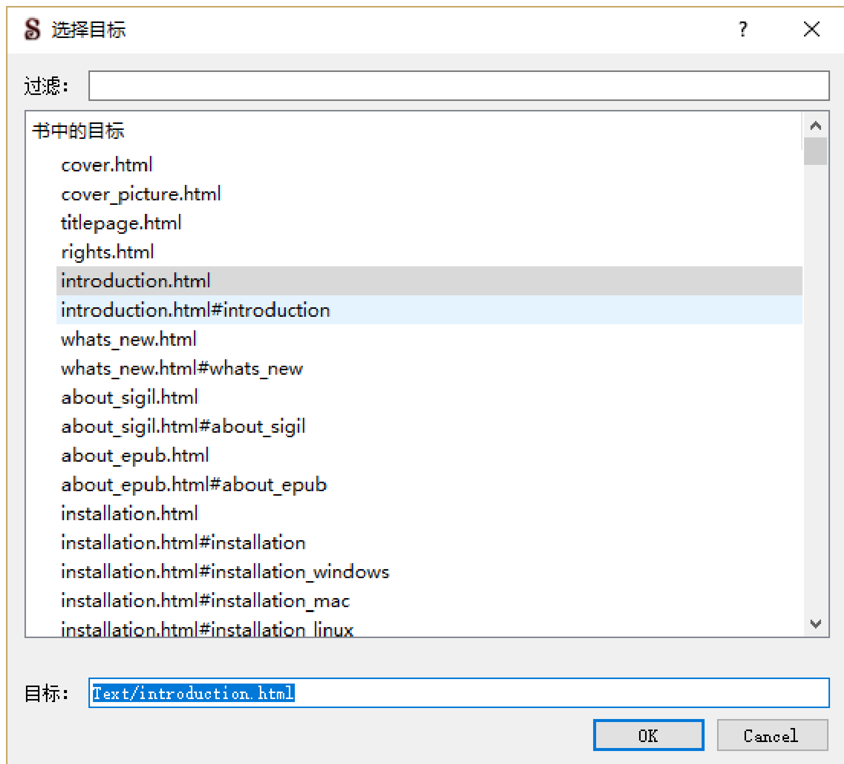


图5-38　选择目录的链接目标

右键菜单允许您删除、重命名条目或全部展开、全部折叠列表。

> 目录条目必须指向有效的目标。如果您只想在目录中找到一个标题，请将其指向下一个有效条目。

创建HTML目录

您可以使用菜单项「工具→目录→创建HTML目录」来为您的ePub创建内联HTML目录。



图5-39　网页目录

内联目录只是ePub中的另一个HTML页面，其中包含文本格式的目录，可以用于跳转到特定章节的链接。

对于大多数ePub电子阅读器，直接显示存储在「toc.ncx」中的目录（「生成目录」生成的目录）。然而，有时候在ePub中需要（或者暂时需要）内联目录，因为ePub将被转换为不使用ePub目录文件的另一种电子书格式。

菜单项「创建HTML目录」将覆盖任何现有的HTML目录文件（如果它具有目录语义）。如果没有这样的文件，它将创建一个名为*TOC.xhtml*的HTML文件，并将其语义类型标记为目录。如果*TOC.xhtml*已经存在，它将被替换。

> 创建目录后，您可以更改内联目录的内容，但是请记住每次创建目录时都要执行此操作。

自定义HTML目录样式表

「创建HTML目录」生成的页面使用单独的CSS文件（sgc-toc.css）格式化目录。如果文件已经存在于您的书中，则不会被覆盖（因此您可以自定义该文件而不会丢失更改）。但是如果不存在，它将在您的「首选项文件夹」中查找*sgc-toc.css*并将其用作默认文件。如果您的「首选项文件夹」中没有文件，则Sigil将为目录创建一个默认样式表。

5.3.11　验证：确保您的ePub是无错误的

为了被认为是有效的ePub，您的电子书必须符合为ePub定义的标准。这意味着它必须包含某些文件和信息，而不包含ePub规范定义的任何错误。为了帮助您检查您的ePub是否有效，您可以使用以下工具。

使用FlightCrew进行ePub验证

选择菜单项「工具→Well-Foremd Check EPUB」或键盘快捷键。

验证结果

文件	行	Offset	错误信息
spellcheck.html	39	N/A	Improperly nested tags: parsing end tag "p" but current parse path is "html.body.div.p.a". See line 39 col 24. near column 106

图5-40　验证信息

Well-Foremd Check EPUB作为Sigil的一部分，它将在您的编辑窗口的下方显示一个新窗口，列出发现的任何问题。生成报告后，查看列出的每个问题，然后编辑文档修改它们。

> 您可以双击大部分相关条目进入该书报错的位置。

您应该每次更改文档后进行验证，以确保没有错误产生。

常见验证问题：

> 请注意，错误以粉红色突出显示，警告以黄色突出显示。

- 该资源存在于OPF `<manifest>`中，但不可访问（未使用）。
  - 此警告意味着列出的文件包含在您的ePub中，但您尚未在书中的任何位置使用它。如果你打算使用它，你应该链接到它，否则你可以删除该文件。尝试搜索所有文件或toc.ncx的文件名——您应该在搜索时删除路径名，例如 如果错误在Text/Chapter1.html中，那么只需搜索Chapter1.html
- 这个OPS文档是可达的，但不存在于OPF `<manifest>`或`<spine>`中。「可达」意味着在epub中存在指向该资源的某种引用。
  - 该错误表示您有一个参考或链接到ePub中的文件或图像，但该文件不包括在ePub中。您需要将文件添加到ePub或更正文本中使用的文件的名称。通常可以通过搜索所有HTML文件中列出的文件名来找到这些。您可能还需要重新生成或更新您的目录。
- ID值『SOMEIDNAME』不唯一
  - 这个错误意味着您的ePub中有一个名为SOMEIDNAME的ID被定义不止一次。

通常这些是头文件或脚注，可能是由复制和粘贴或合并引起的。您可以通过搜索所有HTML文件找到列出的ID名称并更改它们，以便所有ID与所有其他ID不同。如果问题出现在标题中，并且您已自动生成标题，则可以删除ID，并在下次创建目录时重新生成ID。

- 元素『SOMEELEMENT』未声明属性『INVALIDATTRIBUTE』
  - 此错误意味着您有一个名为SOMEELEMENT的HTML标记，其中包含属性INVALIDATTRIBUTE，但INVALIDATTRIBUTE对该标记无效。例如＜h2 invalidattribute＝"test"＞。这可能是由打字错误或旧版本的HTML造成的。您需要在所有HTML文件中搜索INVALIDATTRIBUTE，并删除，更正或替换每次出现。
- 没有发现元素『INVALIDELEMENT』的声明
  - 此错误表示您有一个名为INVALIDELEMENT的HTML标签尚未定义。通常这只是因为它是无效的，是需要更新的旧HTML语法。您需要在所有HTML文件中搜索INVALIDELEMENT，并使用有效的HTML删除，更正或替换它们。
- 内容模型不允许使用元素『INVALIDELEMENT』（p｜d｜h2｜h3｜h4｜h5｜h6｜div｜ul｜ol｜...
  - 这个错误意味着你有一个名为INVALIDELEMENT的HTML标签，对于ePub来说是无效的HTML。通常这是需要更新的旧HTML语法（例如应该使用样式实现的`<center>`标签）。您需要在所有HTML文件中搜索INVALIDELEMENT，并使用有效的HTML删除，更正或替换每次出现。
- 值『SOMENAME』无效『NCName』
  - 此错误表示您的ID名称无效。它可能包含空格或其他无效字符。您需要搜索SOMENAME并进行更改。

验证样式表

您还可以使用W3C网站轻松地验证CSS样式表文件，方法是选择菜单项「工具→验证样式表」。

这将把您所有样式表的内容发送到W3C服务器中并生成问题报告。

您可以通过右键单击「图书浏览器」中的样式表并选择「用W3C验证」选项来验证单个样式表。

> 使用CSS　2.1进行验证。这是接近但不完全是ePub读者使用的版本——所以你应该总是检查你的样式效果在实际的电子阅读器中。

ePub验证与EpubCheck

作为附加检查，您还应该通过EpubCheck程序运行您的ePub。您可以在线使用或下载本地版本的程序：

- IDPF EPUB Validator

ePub验证与设备

如果您使用不了：那么应该尽可能多地在ePub阅读设备或程序中测试实际显示效果。

5.3.12 拼写检查：检查和纠正拼写

Sigil提供了几种方法来检查您的拼写——突出显示拼写错误的单词，一次查找一个单词，或一次显示所有拼写错误的单词。您可以选择使用标准字典，并用您自己的词汇创建个人用户字典。

> 「拼写检查」需要使用「代码视图」来显示拼写错误的单词。

配置拼写检查

您可以设置所需的拼写检查首选项——包括要使用的主要字典以及使用哪些个人用户字典。您可以为不同类型的书籍或个人书籍设置用户词典，或仅使用默认用户字典。您可以随时更改用户字典。

> 在首选项中设置您的字典。

自动拼写检查

在「首选项」或「工具→拼写检查→突出拼错的单词」中启用自动拼写检查时，任何拼写错误的单词在「代码视图」中将以红色波浪下划线突出显示：

```
The Book Browser organises your files into several folders
```

然后，您可以右键单击拼写错误的单词来列出选项列表：

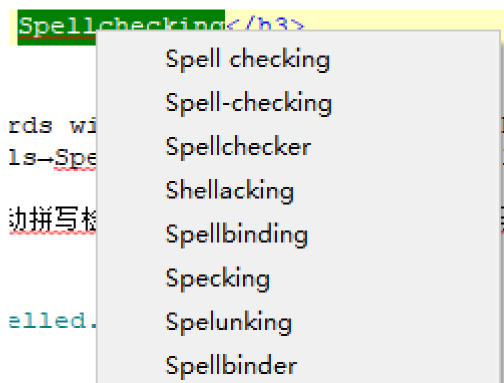


图5-41　拼写错误

- 建议单词：可供选择的替代单词列表——选择一个将替换拼写错误的单词。
- 添加到默认字典：将此单词添加到您当前的默认用户字典。

- 添加到字典：将此单词添加到您从列表中选择的词典中。
- 忽略：忽略整个文档中该单词的拼写，直到重新启动Sigil。

寻找拼写错误的单词

在「代码视图」中找到下一个拼写错误的单词，方法是选择菜单项「工具→拼写检查→下一个拼写错误」或使用键盘快捷键方式。如果你没有进入「代码视图」，执行「下一个拼写错误」功能后，会自动切换到「代码视图」，每次搜索下一个单词时，下一个拼写错误的单词会突出显示。

为了可以从建议选项中选择单词替换拼写错误的单词，或将其添加到字典中，您必须打开「工具→拼写检查→突出拼错的单词」功能。

拼写检查

要打开「拼写检查」对话框，请单击「拼写检查」按钮，或使用菜单项「工具→拼写检查→拼写检查」。

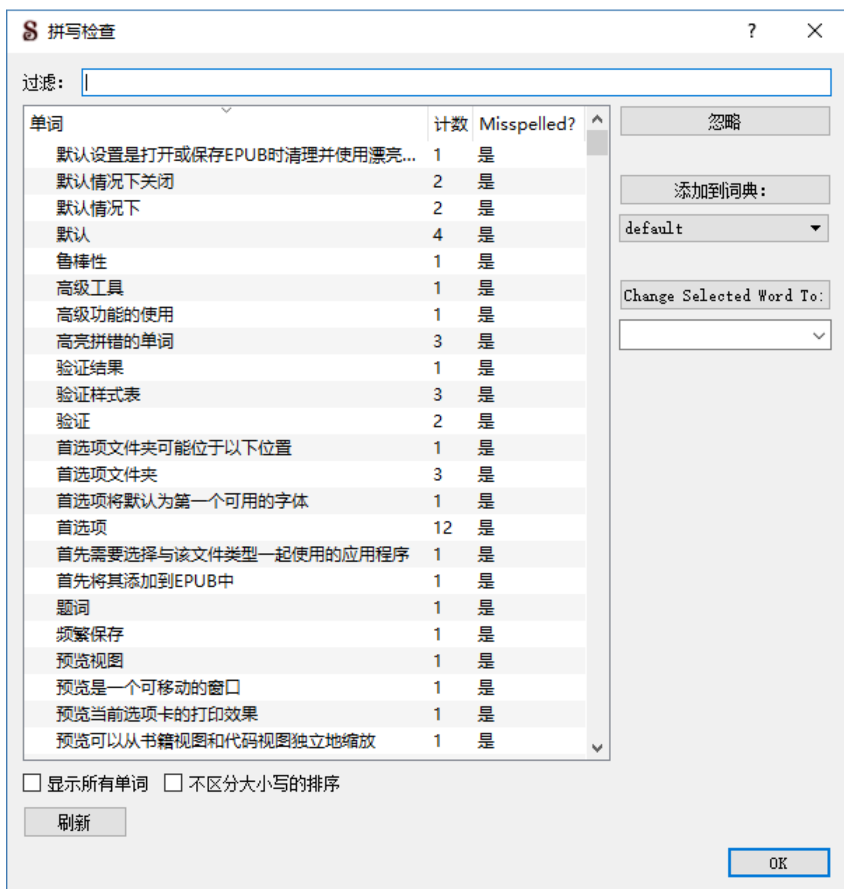


图5-42　拼写检查

在书中显示单词出现的次数，如果它们拼写错误。选择一个单词来执行一个动作。

> 点击单词会在「代码视图」中显示该单词的第一次出现的位置，以便您可以查看其上下文，并根据需要来更改。
> 双击所选单词，查找该单词的下一个拼写错误。

您可以通过单击添加到词典，将单词添加到字典。从下拉列表中选择您的用户字典以添加单词：

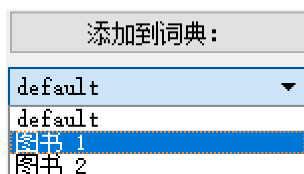


图5-43　填写单词到词典中

要忽略单词，请单击忽略。只有在「首选项」中启用字典时，拼写错误的单词列将更新为「否」。要清除所有被忽略的字，使用「工具→拼写检查→清除拼错的单词」。当您打开新书时，也会清除被忽略的单词。

要更正单词的拼写，请从下面的下拉列表中选择一个建议的单词，然后单击「Change Selected Word To:」或在框中键入自己的单词。然后单击「Change Selected Word To:」。这会将所选词的全部拼写错误更改为新词。它还将自动刷新HTML文件中的单词列表。

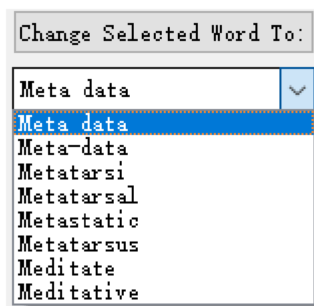


图5-44　更正拼写错误的单词

您可以通过选择「显示所有单词」来查看ePub中的所有单词。此外，列名称的工具提示显示您的ePub中的单词数：

单词	计数	Misspelled?
默…理并使用漂亮...	1	是
默…	2	是
默…	2	是
默认	4	是

Misspelled Words: 3141
Total Unique Words: 5523

图5-45　查看单词书目

如果启用「不区分大小写的排序」，则这些单词将被排序为AaBbCc而不是ABCabc。

如果您在拼写检查打开时编辑书籍，请使用刷新按钮。如果您只是忽略、添加或更改单词，则不需要使用它。

5.3.13 图像、视频、音频：将媒体文件插入到书中

您可以将任何图像、视频或音频文件插入到书籍中，单击「插入文件」按钮或选择菜单项「插入→文件」或键盘快捷键方式。如果文件尚未添加到您的ePub中，您可以从您的计算机中选择它。

> ePub 2和大多数电子墨水阅读器一样只支持图像，但是如果需要，可以插入视频和音频。

将媒体文件插入到书籍中时，首先将其添加到ePub中，然后创建一个链接。这允许您添加一次文件，然后再次链接到它，或者一次添加大量文件，然后再创建一个或多个链接。

将媒体文件添加到您的ePub中

您可以通过以下几种方式将图像、视频或音频文件添加到书籍中：

- 单击「添加现有文件」按钮。
- 选择菜单项「文件→添加→现有文件」。
- 右键单击「图书浏览器」中的文件，然后选择「添加现有文件」。

所有这些将显示一个标准文件对话框，允许您从计算机中选择一个或多个文件。选择任意数量的媒体文件。如果书中已经存在该文件，系统将提示您是否需要替换它们。

这些文件将根据文件扩展名添加到「图书浏览器」中的相应文件夹中。但是，它们还没有在书中的任何地方使用——它们只是可以使用。要使用它们，您需要链接到或插入到书中。

您实际上可以在一个步骤中将文件添加到书籍中并将其插入书中：

- 单击「插入文件」按钮，然后单击其他文件。
- 在「书籍视图」中右键单击文档，然后选择「插入文件」，然后单击其他文件。
- 选择菜单项「插入→文件」，然后单击其他文件。

将文件插入到书中

要将图像、视频或音频文件插入到书籍中，请单击插入图像文件的位置（在「书签视图」或「代码视图」中）。然后单击「插入文件」按钮：

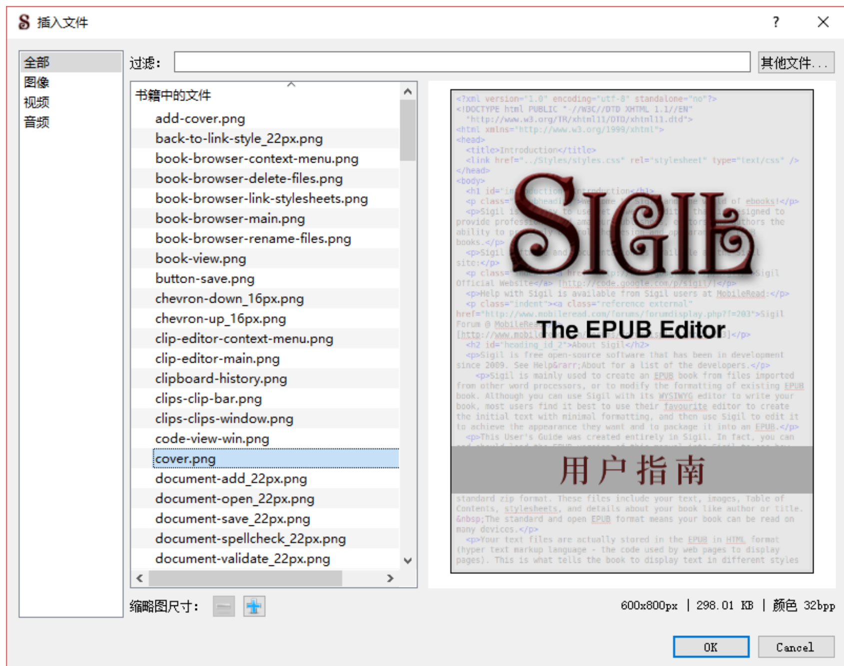


图5-46　插入文件

这将显示您的书中的文件列表。选择要插入的一个或多个文件，然后单击OK，或者双击所需文件，以便自动选择并插入。

> 您可以通过单击其他文件按钮从计算机而不是ePub中选择文件。文件将被添加到您的书籍中，然后自动插入。

您可以通过选择左侧的文件类型或在「过滤」文本框中输入文件名的一部分来显示过滤后的文件。

您可以通过单击左下角的缩放按钮来选择缩略图的大小，从而为「插入文件」对话框中的图像启用缩略图。启用了缩略图的对话框访问较慢，因此您可以将其缩小，直到它们消失为止。

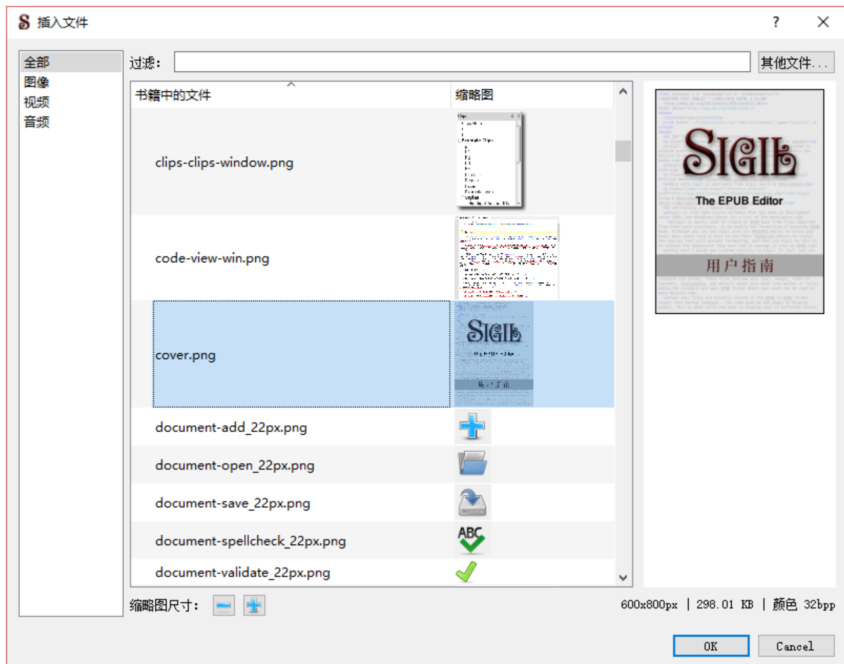


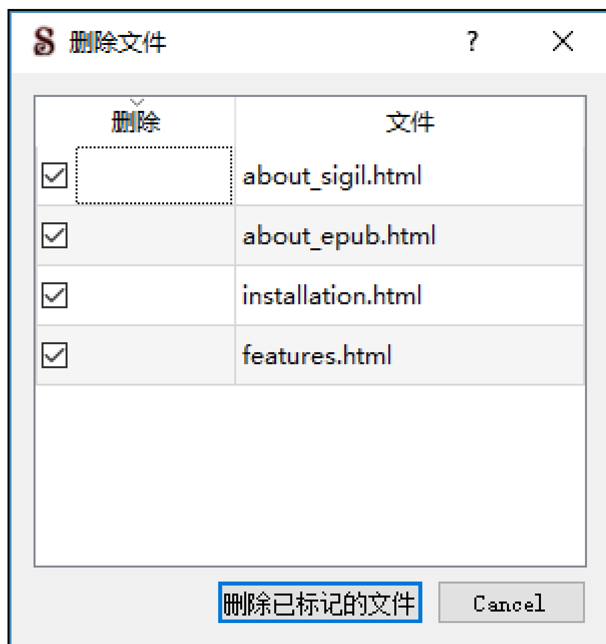
图5-47　插入文件的图片缩略图

> 您还可以在「书籍视图」中右键单击，然后从右键菜单中选择「插入文件」。

查看文件详细信息

要查看图片、视频或音频文件的详细信息，在「图书浏览器」中双击其文件名，或在「书籍视图」中右键单击，然后选择「打开『选项卡』」。

对于图像来说，将显示宽度和高度（以像素为单位）、文件大小（KB）以及图像的颜色像素深度：



---

300 × 320px | 376.02 KB | 颜色 32bpp

图5-48　图片详细信息

编辑媒体文件

Sigil允许您在ePub中轻松编辑图像、视频或音频文件。

只需右键单击「图书浏览器」中的图像，或在「书籍视图」中右键单击其文件，然后选择「打开方式」菜单。然后再选择相应的外部编辑器打开文件。

有关详细信息，请参阅外部编辑器一章。

删除未使用的媒体文件

您可以选择菜单项「工具→删除未使用的媒体文件」，以自动删除任何未使用的图像、视频或音频文件。Sigil会向您显示一个未链接HTML或CSS的文件列表，然后允许您确认要删除哪些文件——或者取消删除。

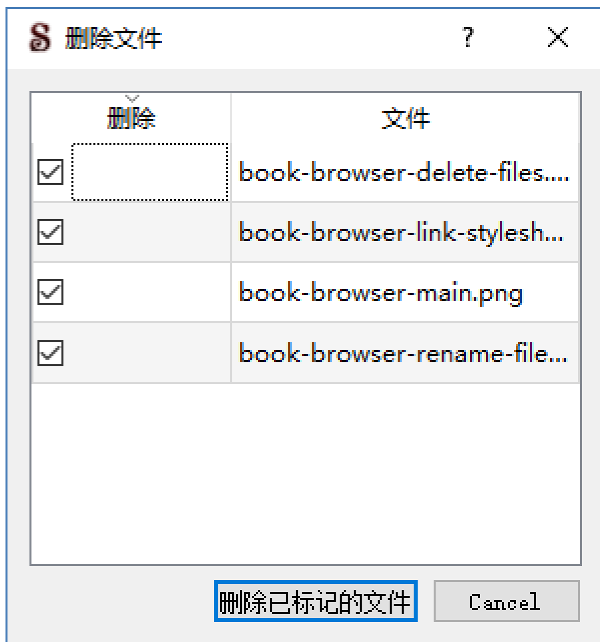


图5-49　删除未使用的媒体文件

为了更好地控制删除单个文件，您可以使用「报告」查看文件的使用次数，然后确认删除所需文件的次数——请参阅报告一章。

> 您可以双击文件以在选项卡中显示该文件。

封面图片

您可以使用添加封面快速添加封面图像。

但是，如果要手动添加封面，请注意，封面的图像是特别的，您必须告诉ePub应使用哪个图像作为封面。不同的电子阅读器使用不同的方法来识别封面，因此您应该使用以下两种方法：

- 在书的开始创建一个「cover.html」文件，然后将图像插入到其中。
- 通过在「图书浏览器」Text文件夹中右键单击cover.xhtml文件作为封面文件，选择「添加语义」并选择「封面」。
- 通过在「图书浏览器」Images文件夹中右键单击封面文件，选择「添加语义」并选择「封面」，将您的实际封面图像文件标记为封面。

> 请确保您有权使用图书中的任何图像，因为大多数图像均受版权保护，其使用可能需要授权。

5.3.14 特殊字符：插入特殊字符

您可以使用「插入特殊字符」对话框将可能不在键盘上的字符插入到文档中。

要插入特殊字符，请单击「插入特殊字符」按钮Ω，或选择菜单项「插入→特殊字符」。

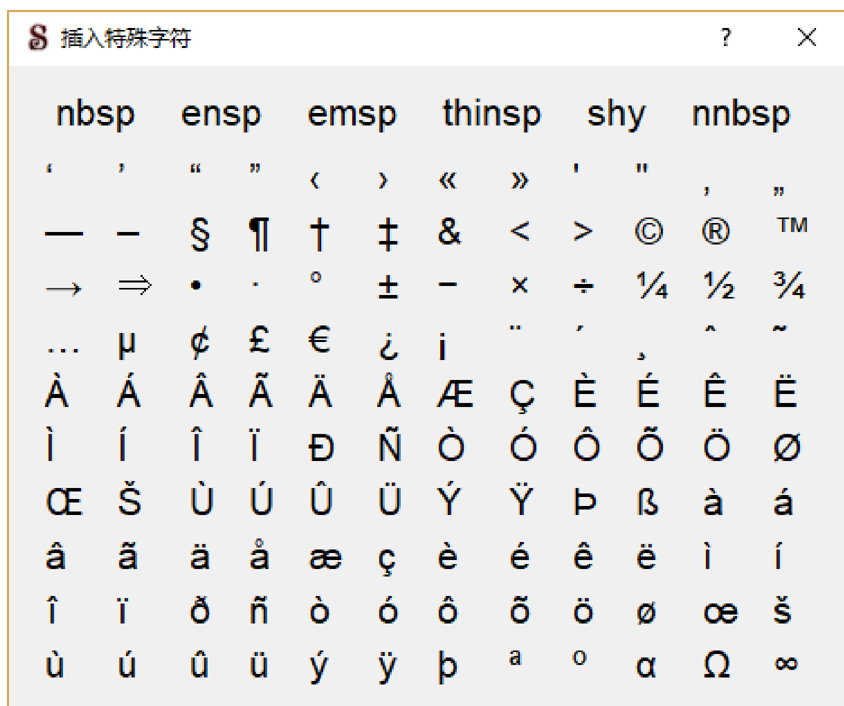


图5-50　插入特殊字符

对话框将保持打开状态，允许您多次插入文本，您可以将窗口移动到所需屏幕上的任意位置。

> 按住Ctrl键并单击选择要插入的字符，完成后将立即关闭对话窗口。

如果你把鼠标停留在字符上，会出现一个工具提示，列出该符号的HTML的字符实体和描述：

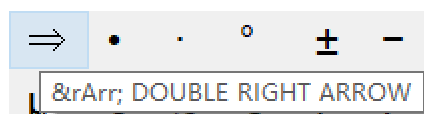


图5-51　特殊字符提示

您可以使用首选项设置对话框中字符的字体和大小。

某些字体可能无法正确显示某些字符。

> 您还可以为您经常使用的特殊字符创建剪辑栏按钮和快捷键。

5.3.15　拆分与合并：操作章节和文件

ePub可以用一个大型HTML文件包含所有文本，或者把书籍拆分为多个HTML，每个HTML包含一部分内容。在大多数ePub中，每个主要章节都存储在ePub的HTML文件中。这不是必需的，但它让处理一本书更加容易，并可以使文本更小，阅读和编辑更快。您可以使用「图书浏览器」访问所有文件。

如果您将一个HTML文件添加到Sigil中，并将其拆分成单独的文件，那么可以使用Sigil的「拆分」功能。如果您有几个较小的文件要合并成较大的文件，则可以使用「合并」功能。

> 大多数电子阅读器在新页面中加载每个新的HTML文件——如果您将章节分割成单独的文件，则可以在章节之间自动分页。

拆分单个文件

您可以通过单击「在光标处拆分」按钮或选择菜单项「编辑→在光标处拆分」，在光标位置将文件分割为2个文件。

这是拆分单个文件的最简单方法。只需在您的文档中确定要分割文件的位置，然后单击，例如在章节名称的开头：

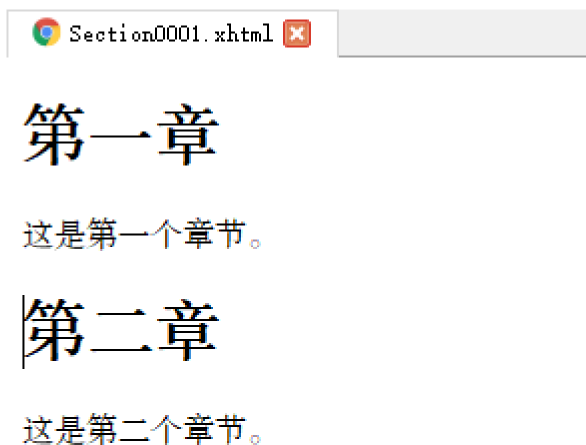


图5-52　在拆分处插入光标

然后点击，分割文件：

Section0001.xhtml Section0002.xhtml

第二章

这是第二个章节。

图5-54　光标后的文件

从原始文件开始到光标位置的所有内容都保存在第一个文件中，点击之后的所有内容都将放入新文件中。请注意，第一个文件保留其名称，第二个文件创建了一个新名称来保存文件。您可以根据需要，重新命名和编辑文件。

拆分多个文件

如果要将多个文件或大文件分割成多个文件，则可以使用添加拆分标记的方法。

1.滚动浏览您的文件，无在你想要一个新的章节时，开始点击文档来设置你的光标位置。然后选择菜单项「插入→拆分标记」或键盘快捷键方式。这将在您的文档中插入代码，告诉Sigil你想要从这里开始一个新的章节——您将在书查看中看到一个水平线，确切地说，您插入标记的位置：



图5-55　拆入拆分标记

2.标记所有章节后，使用「编辑→在章节标记处拆分」。这将自动将您

的所有文件分成与标记的文件数量相同的文件。如果您决定不拆分文件，可以删除标记代码。

一旦您拆分文件，您可以重命名它们。而且甚至可以将其中的一些或全部合并在一起。

> 如果您有很多章节，您可以使用「查找&替换」在每个章节标题之前插入拆分标记。如果您拆分文件，任何标题信息都将复制到新文件中。

合并文件

如果要将两个或多个文件合并到一个文件中，可以使用「合并」功能。

要合并两个相邻文件，请右键单击第二个文件，然后选择菜单项「合并」，或选择第二个文件并使用合并键盘快捷键方式。第二个文件将被合并到第一个文件（上面的文件）中。不会提示您确认合并。

要合并两个以上的文件，请选择要合并的所有文件，然后右键单击选择「合并」。系统将提示您确认合并：

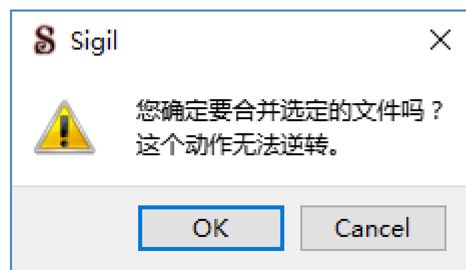


图5-56　合并文件

一旦您点击「OK」，您的文件将被合并到你选择的第一个文件中。

> 在拆分或合并之前保存您的ePub！

5.3.16 「查找&替换」：搜索代码视图

查找和修改文本是Sigil中最常见的操作之一——查找单词、删除一个文件中不需要的文本或更改整本书中特定项目的格式。事实上，如果您需要对文本进行大量更改，您几乎肯定会使用「查找&替换」（及其相关的「搜索模板」功能），因为它允许您对所有文件进行快速大量的编辑。

> 有关高级搜索的信息，请参阅正则表达式教程。

要打开「查找&替换」对话框，请选择菜单项「搜索→查找&替换」或其键盘快捷键：

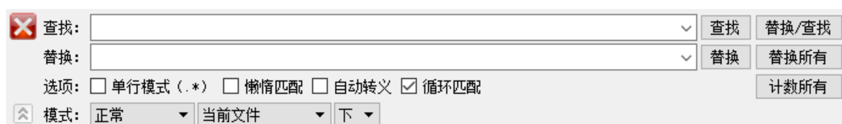


图5-57　「查找&替换」窗口

> 「书籍视图」仅支持在当前文件中查找文本——所有其他操作都将在「代码视图」中完成。

您可以设置各种模式（正常、区分大小写、正则表达式，当前文件、所有HTML文件、选择的文件，向下搜索/向上搜索等）和选项（单行模式、懒惰匹配等），选择操作的类型（查找、替换、替换所有等），例如：

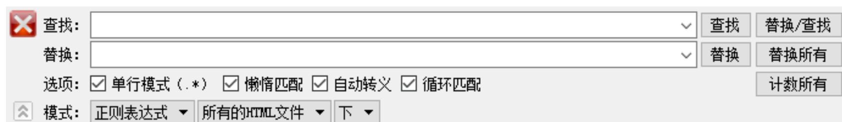


图5-58　「查找&替换」窗口

打开时，为了便于访问，「查找&替换」对话框始终位于编辑窗口的底部。您可以使用左下角的按钮显示或隐藏「选项」设置。

> 「查找&替换」操作的键盘快捷键，如Ctrl＋G表示「查找下一个」。查看「搜索」菜单中的快捷键列表。

操作

右侧的按钮用于运行搜索命令。

> 运行搜索后，您在「查找」或「替换」框中输入的文本会保存，可供您稍后再使用。单击「查找」或「替换」文本框右侧的向下箭头，显示搜索历史记录列表。
> 无论何时运行搜索，结果将显示在「查找&替换」状态栏中。

查找

突出显示查找到的字词。

要搜索文本，请在「查找」文本框中输入文本，然后按查找按钮。如果您正在执行「查找」，「替换」框中的任何文本将被忽略。

> 如果您选择了文档中的文本，您可以使用Ctrl＋F快捷键打开「查找&替换」窗口，并将所选文本填充到「查找」框中。

替换

替换一个由「查找」匹配出的文本。

> 「替换」在「代码视图」中完成，因此您必须首先在「代码视图」中运行「查找」。

替换文字：

- 在「查找」框中输入要查找的文本。
- 在「替换」框中输入要替换的文本。
- 选择查找，搜索并高亮匹配项。
- 选择替换，来进行替换。

> 要撤消更改，您可以使用「撤消」按钮或右键单击编辑窗口，然后选择「撤消」以更改文本。但是，如果要替换文件，而您的下一个匹配位于不同的页面中，则需要在选择「撤消」之前切换到包含替换文本的文件中。

替换/查找

替换下一个匹配项，然后运行「查找」，自动高亮下一个匹配项。

要替换文本，请在「查找」框中输入旧文本，并在「替换」框中输入新文本。然后单击替换/查找。

第一次做一个替换/查找时，第一个匹配将被高亮。随后使用替换/查找将替换高亮的匹配文本，然后自动查找下一个匹配项。

> 如果没有选择文本，您可以安全地使用「替换/查找」，因为它将像「查找」一样执行，直到高亮匹配项。

替换所有

根据您的设置，将匹配「查找」的文本并全部替换为「替换」框中的文本。这一切都是自动完成的。

> 您无法撤消全部替换——所以先保存一个备份！

替换所有始终从文件的开始位置替换。

> 使用「替换所有」之前，先使用「计数所有」统计您的「查找&替换」字符串，以避免出现意外。
> 在「替换所有」之前备份您的书，并在更换后再检查，以避免发生问题。
> 检查您的模式是否设置为当前文件或所有HTML文件

计数所有

「计数所有」可以使用「选项」按钮 显示（见下文）——或者使用键盘快捷键方式。

「计数所有」显示在所选文件中出现「查找」文本框中输入的文本的次数。它使用「上/下」模式和「循环匹配」选项的设置。

> 在进行替换之前，先使用计数所有来检查替换的次数。并且您应该在「替换所有」后运行它，以防特定的搜索和替换出现新的匹配。

模式：搜索内容

您可以使用这些选项来控制要搜索的文本类型。

正常

此模式将搜索您所输入的内容，但它将忽略大小写，例如 「A」和「a」将被视为同一单词。

区分大小写

此模式将搜索您正在输入的内容。

正则表达式

正则表达式是「规则表达式」的缩写。此模式允许您搜索特定的规则文本，而不是单词或字符的字面值。正则表达式非常强大，是使用Sigil处理电子书的主要功能之一。

例如，您可以搜索「Page[0-9]+」，其中将找到「Page」，后跟一个或多个数字。

> 有关使用正则表达式的更多详细信息，请参阅「正则表达式」一章。

模式：搜索范围

您可以使用这些选项来控制要搜索哪些文件，以便轻松地搜索所有文件或选择的文件。

当前文件

这将您的搜索限制在当前的文件中。您可以搜索HTML文件和样式文件。

如果您的搜索到达文件的末尾，并且设置了「循环匹配」，将会从文件的另一端继续搜索。您将在屏幕上看到一个闪烁的箭头小图片，以指示搜索的范围。

> 如果在按住Ctrl键的同时单击某个操作，您将只能在当前文件上临时运行搜索。

所有HTML文件

这将搜索每个HTML文件。它只会搜索每个HTML文件。

当搜索到达书的结尾或开始时，它将自动移动到下一个或上一个文件（按「图书浏览器」中列出的顺序），如果打开了所有文件，将会循环搜索所有文件。

选择的文件

这只会搜索您在「图书浏览器」中选择的HTML文件。

当搜索到达文件的末尾时，它将自动移动到下一个或上一个选定的文件（按「图书浏览器」中列出的顺序），如果打开了「循环匹配」，将循环搜索所有文件。

标记文本

这是一个特殊选项，仅在标识文本时显示。

您可以通过选择标记文本来限制搜索的文本。要标记文本，请在代码视图中选择一些文本，然后使用「搜索→标识选定文本」或使用右键单击，从菜单项选择「标识选定文本」。

可以通过再次使用菜单项而不选择文本来取消标记。

如果您使用「撤消」、键入任何文本或更改视图或切换选项卡，文本将被取消标记。如果您只是使用「替换」或「全部替换」，标记的文本将

被适当扩展或缩小。

模式：搜索方向

您可以使用这些选项来控制搜索的方向。

向上

从您在文档中的位置向上搜索并查找上一个匹配项。

向下

从您在文档中的位置向下搜索并查找下一个匹配项。

选项

使用「正则表达式」模式时，「查找&替换」提供了几个选项。

单击「查找&替换」窗口上的显示按钮展开窗口（如果尚未展开）以显示「选项」和计数所有按钮：

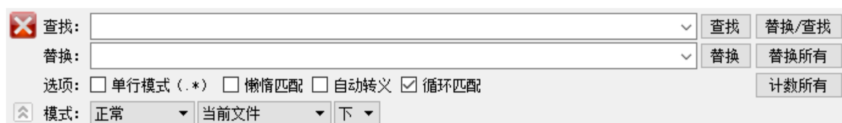


图5-59　显示「计数所有」

点击隐藏按钮隐藏「选项」：

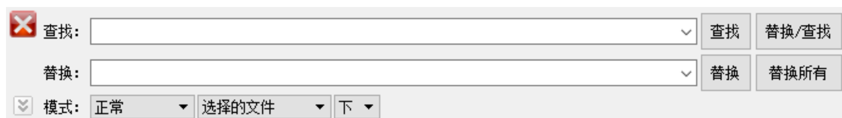


图5-60　隐藏「计数所有」

单行模式

这个正则表达式选项会预处理所有正则表达式搜索——添加前缀`(?s)`，并且当您希望在「.*」模式跨行匹配任何字符时使用。

懒惰匹配

这个正则表达式选项在所有正则表达式搜索中都加上`(?U)`，当您希望匹配最短而不是最长的匹配时使用。

自动转义

先选择文本然后使用Ctrl＋F将文本添加到「查找&替换」时，将空格转换为\s并转义某些字符，以使其适合正则表达式。

循环匹配

启用后，当前文件中的搜索将自动从文件底部到顶部查找文本（反之亦然，如果是向上搜索）。禁用时，搜索将在文件末尾停止。

当您在所有HTML文件中进行搜索时，如果搜索到达最后一个HTML文件，则启用「循环匹配」时，搜索将从第一个HTML文件中自动继续查找文本（反之亦然）。禁用时，搜索将停止在书中的最后一个文件（或第一个，如果是向上搜索）。

右键菜单

如果您右键单击「查找」和「替换」框，您将看到下面的选项：

- 搜索模板：您可以从搜索模板列表中选择条目并加载到「查找」和「替换」框中。
- **Save Search**：保存「查找&替换」搜索到搜索模板中。
- **Tokenise Selection**：将空格转换为\s并转义某些字符。

5.3.17 搜索模板：保存的「查找&替换」搜索

「搜索模板」对话框允许您保存和编辑所有常见的「查找&替换」搜索。您也可以使用它来运行搜索。

要打开「搜索模板」，请选择菜单项「工具→搜索模板」或键盘快捷键方式：

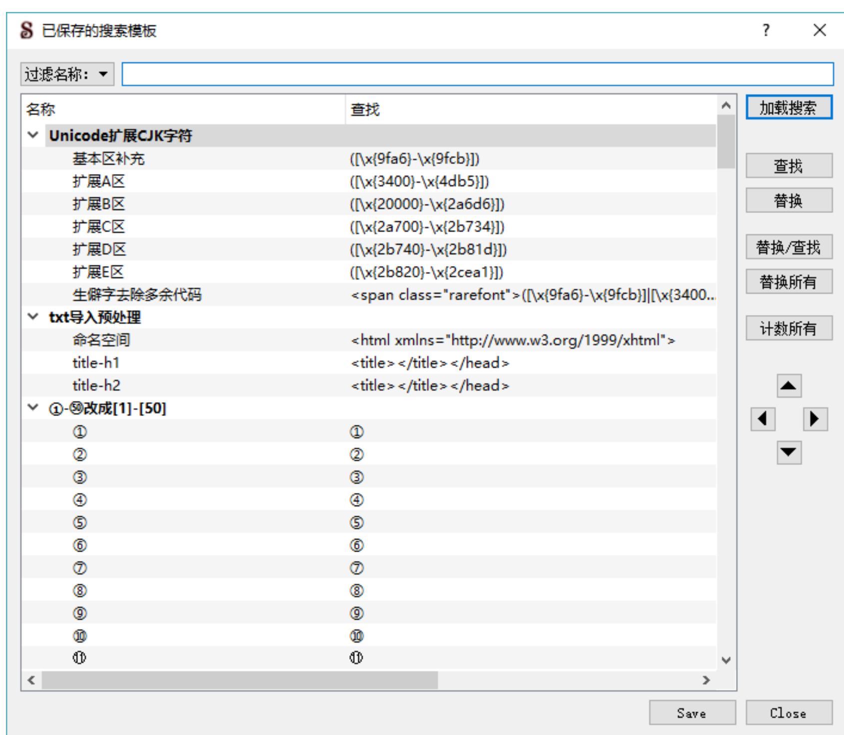


图5-61　已保存的搜索模板

> 所有搜索模板将使用「查找&替换」中设置的选项和模式运行。

右键菜单命令

关于编辑搜索的大多数操作，都可以使用右键单击呼出菜单项：



图5-62　搜索条目的右键菜单

选择一个或多个条目，然后右键单击选择以下选项：

- 添加项目：在您单击的项目下添加空白搜索。
- 添加组：在您单击的条目下添加一个新组。
- 编辑：编辑或重命名条目（与双击相同）。
- 剪切：删除一个或多个可以稍后粘贴的条目。
- 复制：复制一个或多个可以稍后粘贴的条目。
- 粘贴：粘贴从剪切/复制的任何条目。
- 删除：删除条目或组。
- 导入：从文件中读取导出的条目列表到搜索列表中。
- 重新加载：从最后保存的状态中恢复列表。
- 导出：将所选条目保存为文件，以进行共享。
- 导出所有：将所有条目保存到文件中。
- 全部折叠：折叠所有组的显示。
- 全部展开：展开所有组的显示。

运行搜索

「搜索模板」可用于运行搜索以及管理它们。

所有「查找&替换」操作均可从「搜索模板」窗口中获得，您不需要「查找&替换」即可使用它们（尽管在运行搜索时将使用「查找&替换」中设置的选项）。

要运行搜索，只需选择一个条目，然后单击其中一个操作按钮。所有结果消息将显示在「查找&替换」窗口中。

> 虽然是使用「查找&替换」中设置的模式完成所有搜索，但是您可以在选择搜索时按住Ctrl键，强制其在当前文件中临时运行。

组搜索

您实际上可以运行一组搜索。每个搜索将按顺序运行。

只需选择组名称，例如，运行计数所有。或选择几个非组条目运行，例如全部替换。

保存

要将条目保存到计算机中，请单击Save。

要退出而不保存您的条目，您可以关闭窗口（使用Close按钮或Esc键），如果有提示的话，请确认您不想保存。您还可以使用右键菜单「重新加载」从计算机中重新加载条目——这将覆盖未保存的修改。

您可以使用右键菜单项中的「导出」和「导入」来保存搜索，以便与其他人共享。

> 如果您保存并运行Sigil的多个副本，则Sigil的其他实例将自动重新加载新保存的搜索，以使所有副本保持同步。

过滤

您可以使用「过滤器」文本框来限制显示的搜索列表，以便更容易找到搜索项目。

加载搜索

单击加载搜索按钮会将您选择的搜索加载到「查找」和「替换」框中。

加载不会运行搜索——它只是填充「查找」和「替换」字段的一种方式。您也可以右键单击「查找」和「替换」文本框，并从右键菜单中选择搜索模板来执行相同的操作。

如果要从「搜索模板」对话框中运行搜索，则无需加载搜索。

> 您可以使用快捷键打开「搜索模板」，在过滤器框中输入搜索名称并按回车键（如果要关闭窗口，则为Esc键）即可加载搜索。

移动条目

您可以使用箭头键或拖放来移动搜索或组。

您必须先创建一个组才可以组合条目。

在「查找&替换」中使用搜索模板

保存搜索

在「查找&替换」中，您可以通过右键单击「查找」或「替换」文本框并选择「Save Search」菜单项来保存搜索。

加载搜索

您还可以通过在「查找」或「替换」文本框中右键单击，选择对应的搜索模板菜单项，将搜索模板加载到「查找」和「替换」中。

加载搜索后，您可以在运行之前更改模式或选项。

5.3.18 剪切板：保存文本和样式

「剪切板」允许您随时保存文本片段并将其粘贴到文档中。它们也可用于在文本周围插入HTML样式标签。您可以使用「剪切板编辑器」创建剪辑，使用剪辑窗口快速选择剪辑，使用「剪辑栏」工具栏按钮选择剪辑（视图→工具栏→剪辑栏），或使用键盘快捷键来访问剪辑。您甚至可以自动从您的CSS样式表填充「剪辑」列表。

剪切板编辑器

要打开「剪切板编辑器」，请选择菜单项「工具→剪切板编辑器」或键盘快捷键方式：

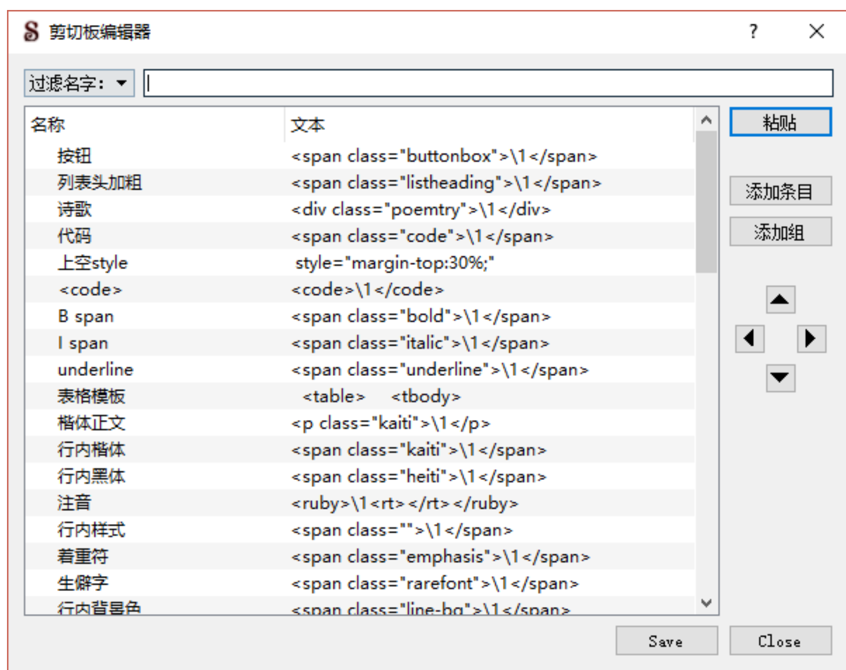


图5-63　剪切板编辑器

右键菜单

关于编辑剪切板的大多数操作，都可以使用右键单击呼出右键菜单来访问：



图5-64　剪切条目的右键菜单

选择一个或多个条目，然后右键单击选择以下选项：

- 添加条目：在您单击的条目下添加空白剪辑。
- 添加组：在您单击的条目下添加一个新组。
- 编辑：编辑条目。
- 剪切：删除一个或多个可以稍后粘贴的条目。
- 复制：复制一个或多个可以稍后粘贴的条目。
- 粘贴：粘贴任何从剪切/复制的条目。
- 删除：删除条目或组。
- 导入：从文件中读取导出的条目到列表中。
- 重新加载：从最后保存的状态恢复列表。
- 导出：将所选条目保存为文件，以进行共享。
- 导出全部：将所有条目保存为一个文件。
- 自动填充：从CSS样式表中创建剪辑。
- 全部折叠：折叠所有组的显示。
- 全部展开：展开所有组的显示。

您也可以使用添加条目或添加组按钮添加新项目。

保存

要将条目保存到计算机中，请单击Save。

要退出而不保存您的条目，您可以关闭窗口（使用Close按钮或Esc键），如果有提示的话，请确认您不想保存。您还可以使用右键菜单「重新加载」从计算机中重新加载条目——这将覆盖任何未保存的更改。

您可以使用右键菜单项中的「导出」和「导入」来保存搜索，以便与其他人共享。

> 如果您保存并运行Sigil的多个副本，则Sigil的其他副本将自动重新加载新保存的剪辑，以使所有副本保持同步。

过滤

您可以使用「过滤器」文本框来限制显示的剪辑列表，以便更容易找到剪辑。

移动条目

您可以使用箭头键或拖动来移动条目或组。

您必须先创建一个组才可以组合条目。

粘贴剪辑

单击剪辑板中的剪辑按钮，将粘贴选择条目到您选择的最后一个窗口中（您的文档在「书籍视图」或「代码视图」或「查找&替换」）。

> 您可以使用快捷键打开「剪切板编辑器」，在过滤框中键入剪辑名称并返回（然后如果要关闭窗口，则可以使用Esc键）粘贴剪辑。但使用键盘快捷键可能会更容易。

如果需要，您还可以选择粘贴组或一个剪切项目。

您还可以右键单击「书籍视图」或「代码视图」中的文本，以显示右键菜单。然后，您可以选择您的剪辑，并将其立即插入到文中。

自动填充CSS条目

自动填充右键菜单项将扫描您的CSS样式表，并为每个找到的简单选择器创建一个剪辑条目。

例如，如果您有选择器或样式「p.first」，则自动填充将为其创建一个剪辑，如下所示：

```
<p class="first">\1</p>
```

如果选择器为「*.class」，则自动填充将创建「p.class」，「div.class」和「span.class」的条目。

所有新的自动填充剪辑都存储在「自动填充」组中。如果「自动填充」组不存在，将会创建它。创建组后，您可以重命名组，您可以将任何条目移动或复制到任何组。

剪辑栏

「剪辑栏」是可选的工具栏，将显示前20个不在组中的剪辑按钮。您可以通过选择菜单「视图→工具栏→剪辑栏」来显示剪辑栏：

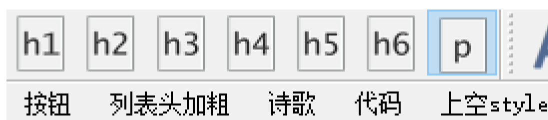


图5-65　「剪辑栏」快捷按钮

按钮将以剪辑的名称来显示，因为您可能希望保留名称。您可以使用它们作为自己样式的快捷键，甚至可以使用常用的特殊字符。

> 剪辑还可以具有键盘快捷键——请参阅「插入→剪辑」

剪切板窗口

「剪切板」窗口是一个可停靠窗口，显示列表中的所有剪辑，并允许您单击条目将其粘贴到文本中。您可以通过选择菜单项「视图→剪切板」来显示「剪切板」窗口：

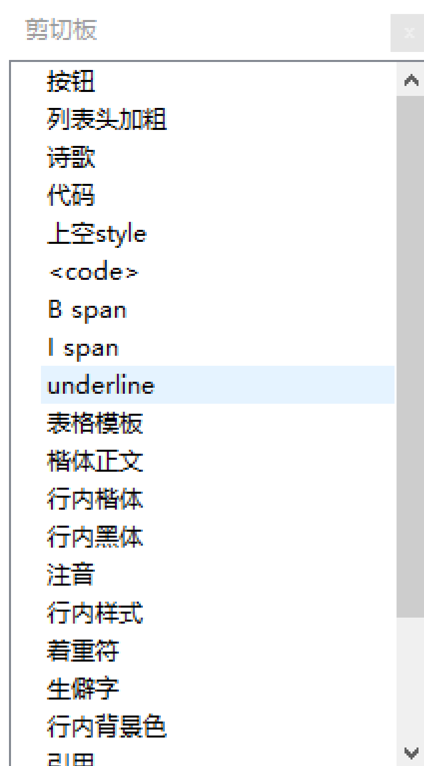


图5-66　「剪切板」窗口

使用剪切板格式化文本

剪切板提供了一种在选定文本之前和之后粘贴文本的方法——为您提供格式化文本的基本功能。

如果您在文本中使用「\1」定义剪辑，则剪辑将其视为正则表达式替换命令。粘贴剪辑时，「\1」将替换为所选文本。

例如，如果你的剪辑是：

```
<span class="name">\1</span>
```

然后选择文字：

```
Hello World
```

当您应用剪辑时，文本将被更改为：

```
<span class="name">Hello World</span>
```

您可以使用它来构建可以快速应用的样式和格式库。任何文本都可以在

代码视图中使用。但是在书籍视图中，特别是跨标签情况下只能使用有限的剪辑。这是因为您将插入到现有的段落中——您不能在段落标签内插入某些标签。

您可以在剪辑中使用其他正则表达式命令，尽管复杂的正则表达式语句可能无法正常工作。

> 粘贴剪辑之前，您可以使用Ctrl＋Space从所选文本中删除格式化标签。

5.3.19　剪贴板历史：剪贴板列表

「剪贴板历史」可以快速访问您最近复制或剪切的文本。

要将文本添加到剪贴板，只需使用「复制」或「剪切」即可。

要从剪贴板历史中国粘贴文本，请选择菜单项「编辑→Edit/Paste From Clipboard History」或键盘快捷键方式：

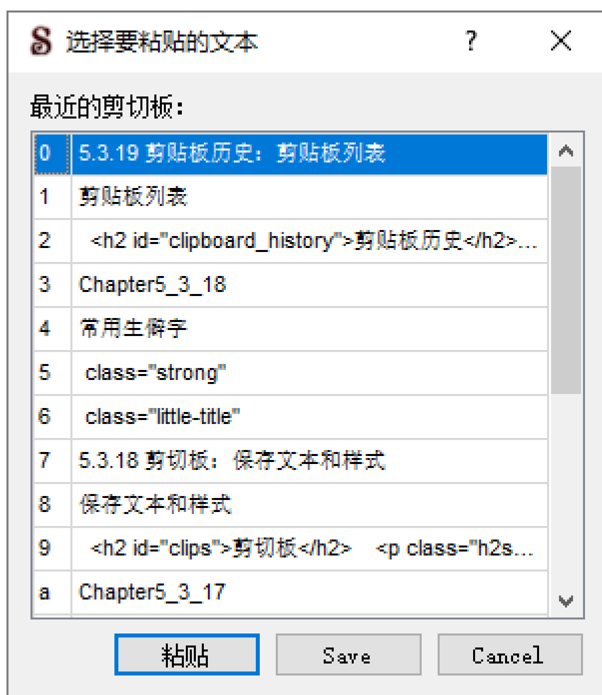


图5-67　「剪切板」历史记录

单击要使用的文本，并将其粘贴到一个选定的窗口中，对话框将自动关闭。

您也可以输入该条目所显示的字符以进行选择。

5.3.20 链接和ID：链接、注释和ID

Sigil提供了一些有用的工具，用于创建脚注、尾注和交叉引用的ID和链接。

插入ID

要将文本标记为链接的目标位置，您需要通过插入锚点ID并为其指定名称。

选择要用ID标记的文本，然后单击按钮或选择菜单项「插入→ID」：

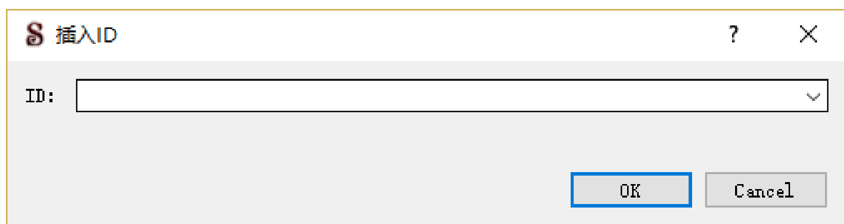


图5-68　插入ID

输入此ID所需的名称，例如footnote-1（必须以字母开头）。

要编辑ID，请选择该单词或单击单词某处，然后再次单击该按钮。

ID实际上是使用如下HTML代码插入的：

```
<a id="footnote-1">Footnote 1</a>
```

> 您不能在同一个文件中使用相同名称的ID。

如果您点击下拉箭头，该文件中的现有ID列表供您参考。

要删除ID，您需要在「代码视图」中编辑HTML代码，以完全删除标签。

> 要识别书籍中的ID，您可以在「代码视图」中进行搜索。或者您可以通过将以下CSS代码临时添加到样式表中，在「书籍视图」中的高亮显示：
>
> `[id class!=sigil_index_marker] { color:#00ff00; }`

插入链接

要创建链接到文档另一部分的链接，请选择要链接的文本，然后单击该

按钮或选择菜单项「插入→链接」：

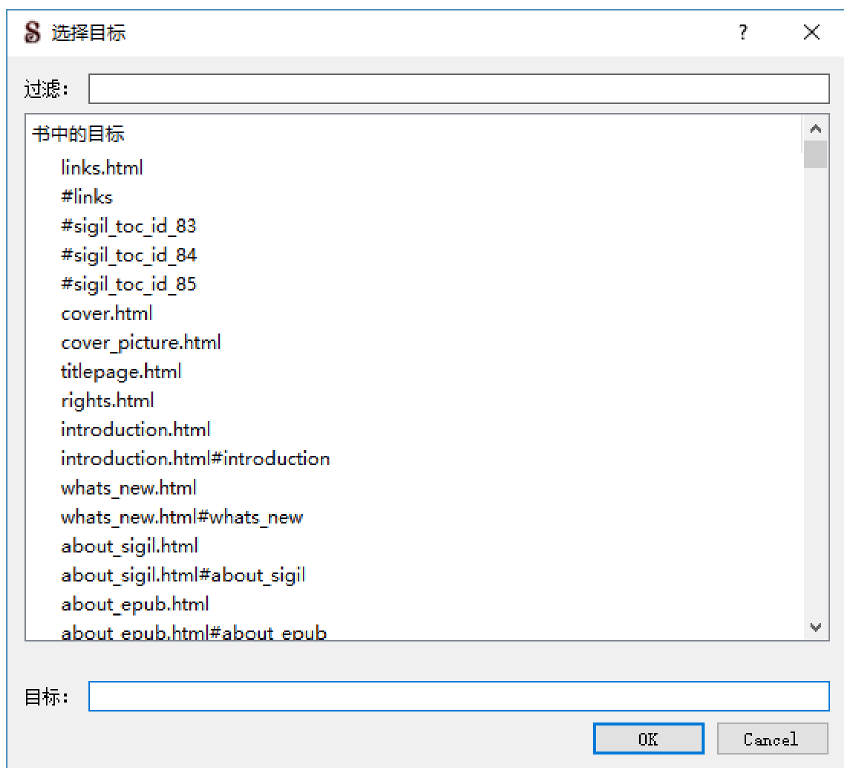


图5-69　选择链接目标

您将获得ePub中有效目标的列表，您可以选择它们（双击一个或选择并单击OK）。ePub中的有效目标是已创建的任何锚ID文件名，甚至可能是图像。列表将被排序，以便首先列出当前文件中的目标，然后按照书顺序列出所有剩余的文件。

要编辑链接，请再次选择文本或使用箭头键将文本移动到文本中（您无法单击它，否则将被带到链接），然后单击按钮。如果它有效，链接将在列表中突出显示。

您还可以通过在「目标」框中键入条目，输入所需的链接目标网址（例如指向网站）。

浏览链接

要浏览「书籍视图」中的链接，只需点击它。

要转到「代码视图」中的链接，请按住Ctrl键，同时单击HTML开始标签。

> 如果文本是链接，则按住Ctrl键单击文本将带您进入链接，否则如果文本只是一个标签，则会将您转到为标签定义的CSS样式。

要返回到您点击链接的位置，只需点击返回按钮（菜单项「搜索→返回」），或选择菜单项「跳转到链接或样式」。

5.3.21　样式：使用样式表

ePub主要是关于样式的编辑。Sigil提供非常易于使用但功能强大的工具来帮助您处理样式表。

样式可以插入到您的单个HTML文件中，但通常的方法是定义包含样式的类，然后将该类链接到HTML文件。

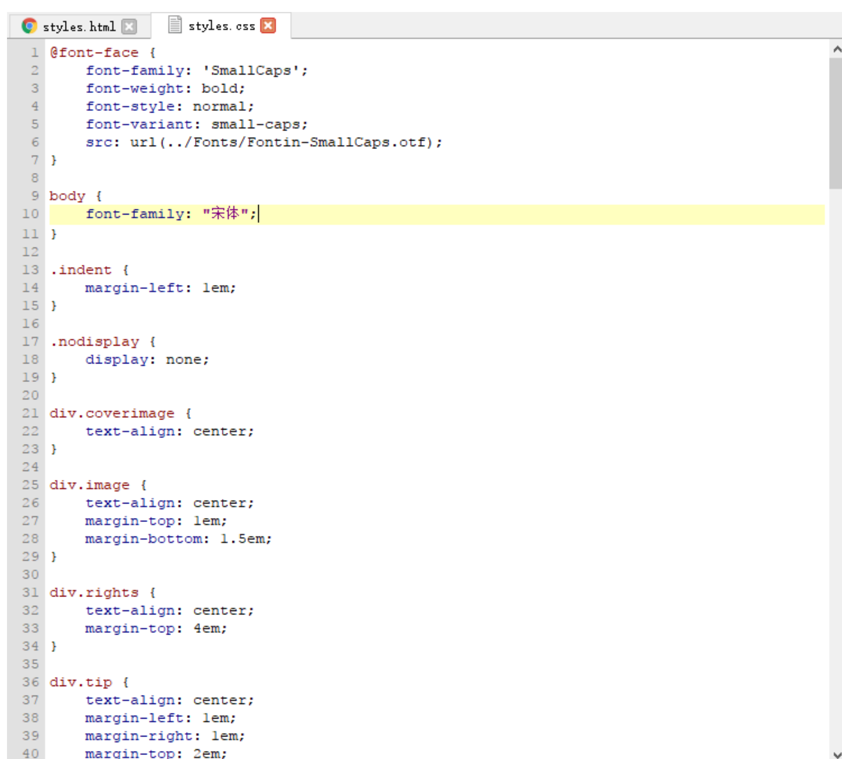


图5-70　样式表界面

创建样式表

要创建样式表，只需使用「文件→添加→空白样式表」。然后只需定义要使用的样式。或者将现有的样式表文件（例如您经常使用的模板样式表）添加到ePub中。

> 不要在样式表文件名中包含空格。

链接样式表

您可以通过在「图书浏览器」中右键单击HTML文件，轻松链接样式

表。选择链接样式表或取消已链接的样式表。

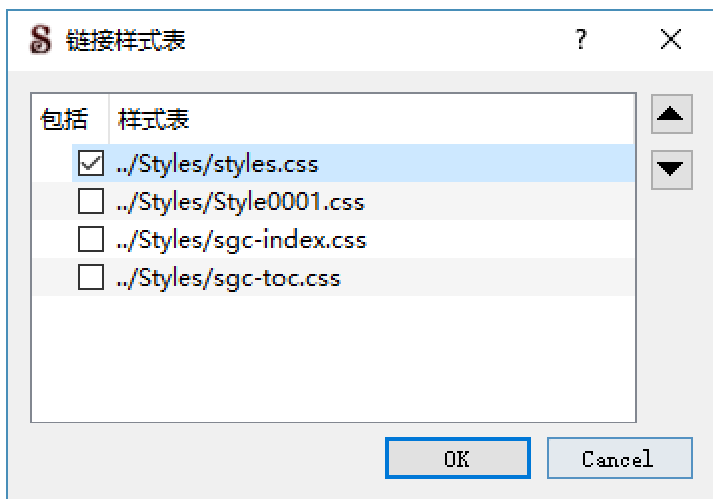


图5-71　链接样式表

浏览样式

您可以快速跳转到HTML代码使用的样式。

在「代码视图」中按住Ctrl键并单击HTML标签，或者选择菜单项「搜索→跳转到链接或样式」或键盘快捷键方式。您将立即跳转到到所选HTML标签的样式确切位置（如果存在）。

要返回到您所在的位置，只需单击返回按钮或选择菜单项「搜索→返回」或键盘快捷键方式。

删除样式

您可以选择菜单项「工具→删除未使用的样式」，以自动删除样式表中的任何未使用的样式。Sigil将向您显示样式表中没有在HTML文件中引用的样式名列表，然后允许您确认或取消删除它们。

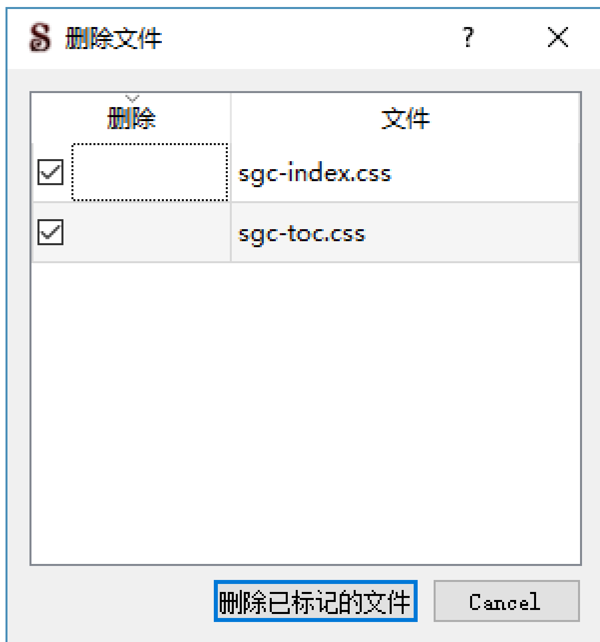


图5-72　删除样式表文件

> 虽然Sigil进行了多次检查以确保样式没在使用，但如果使用了特殊复杂的CSS选择器，则可能会错误地列出样式名，所以您应该仔细检查样式。
> 您可以双击样式以在CSS样式表中显示该样式。

右键菜单和格式化样式

样式表的右键菜单允许您重新格式化样式表：

- 剪切板：选择一个剪辑，添加到样式表中。
- 添加到剪切板：将所选文本添加到剪切板中。
- 跳转到链接或样式：未使用。
- 重新格式化**CSS**
    - 样式多行显示：通过多行排列样式中的声明。
    - 样式单行显示：在一行上排列每种样式的声明。

虽然大多数样式表的格式以多行显示，「样式单行显示」如下所示：

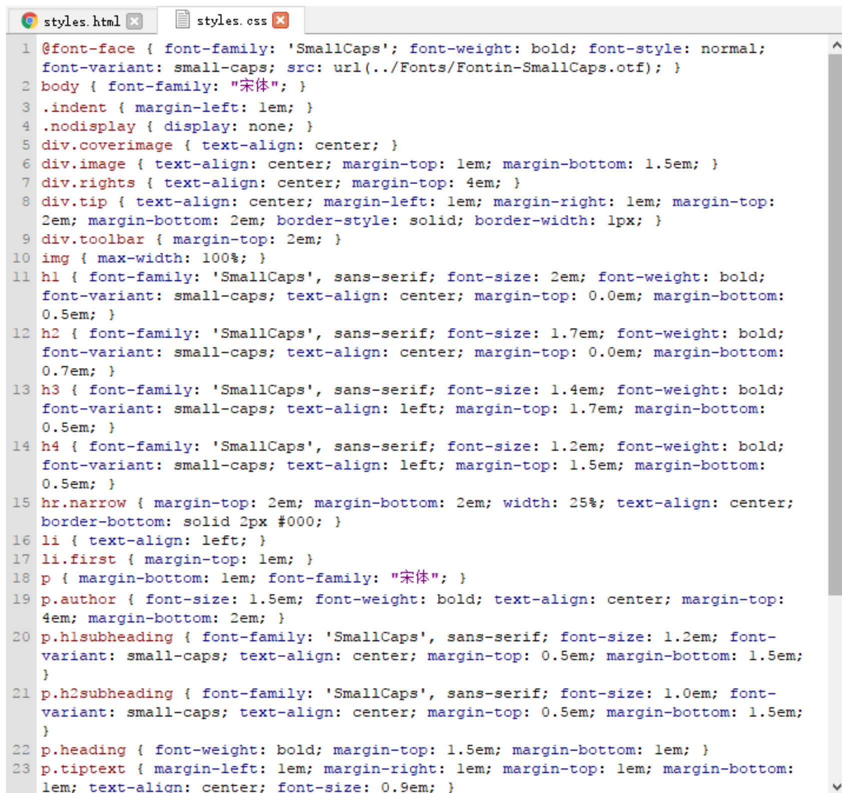


图5-73　样式单行显示

查看样式

您可以使用菜单项「工具→报告」显示有关样式的报告列表——请参阅报告一章。

为了更好地控制删除单个样式，您可以使用「报告」来查看样式的使用次数，然后删除所需的次数。

5.3.22　索引：为您的书创建一个索引

如果要为您的书创建索引，您可以使用Sigil来识别要索引的内容，然后自动创建一个包含索引的HTML页面。

您可以在文本中标记单个单词进行索引，或使用「索引编辑器」确保每次出现的单词都被编入索引中。

标记条目索引

要识别包含在索引中条目的第一种方法是在文本中标记特定的单词或短语。只有您标记的单词或短语才会包含在索引中——如果您需要所有出现的内容，请使用索引编辑器。

高亮「书籍视图」或「代码视图」中的单词或短语，然后选择菜单项「工具→索引→索引标记」或键盘快捷键方式（可以更改）。然后，这将打开一个提示，让您输入如何在索引中显示的单词：

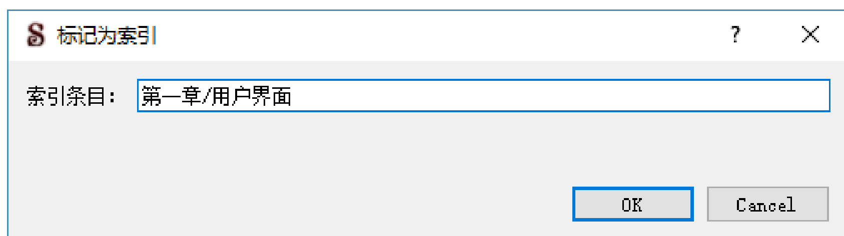


图5-74　标记条目为索引

您在索引中输入的名称将显示在索引中，如下所示：

- [空白]：如果您将该名称留空，则您将在索引中使用高亮的确切字词。
- 单词：如果您输入单词或短语，则将使用该单词或短语（例如Oak而不是oak）。
- 标题/单词：如果您在名称中使用「/」，则将创建一个分层组（例如，Forest/Tree/Oak将创建一个Forest下Tree的条目Oak）。
- 条目;条目：如果你使用「;」那么将为此单词或短语创建多个条目。

当您标记索引文本时，它只是添加一些Sigil特定的HTML代码，告诉Sigil在创建索引时要包括这些单词。在「代码视图」中可能会显示如下：

```
<a class="sigil_index_marker" title="Character/
Alice">Alice</a>
```

如果您不再需要使用Sigil索引的单词或短语，则需要删除该单词的特殊

HTML标签。

> 要查找索引的单词，您可以在「代码视图」中进行搜索。或者您可以通过将以下CSS代码临时添加到样式表中，在「书籍视图」中以高亮的方式显示：
>
> ```
> [class="sigil_index_marker"] { color:#335500; }
> ```

将条目添加到索引编辑器

要识别索引单词的第二种方法是将单词添加到列表中。然后，当您创建索引时，该字词的每次出现将自动包含在索引中。您可以使用为索引标记的单词和单词列表同时进行索引。

在「书籍视图」或「代码视图」中添加特定单词到索引列表中，突出显示单词或短语，然后选择菜单项「工具→索引→添加到索引编辑器」。如果您选择的文本包含任何HTML标签，则应删除它们。

这将弹出索引编辑器，并将该字添加到要索引的条目列表中。您还可以按照下面的说明直接在索引编辑器中输入单词。

创建索引

要创建您的索引，一旦标记了单词或将它们添加到列表中，只需选择菜单项「工具→索引→创建索引」。

这将创建或更新一个名为*index.xhtml*的HTML页面。它将被标记为「索引」语义类型。条目按字母顺序列出，并使用默认样式，如下所示：

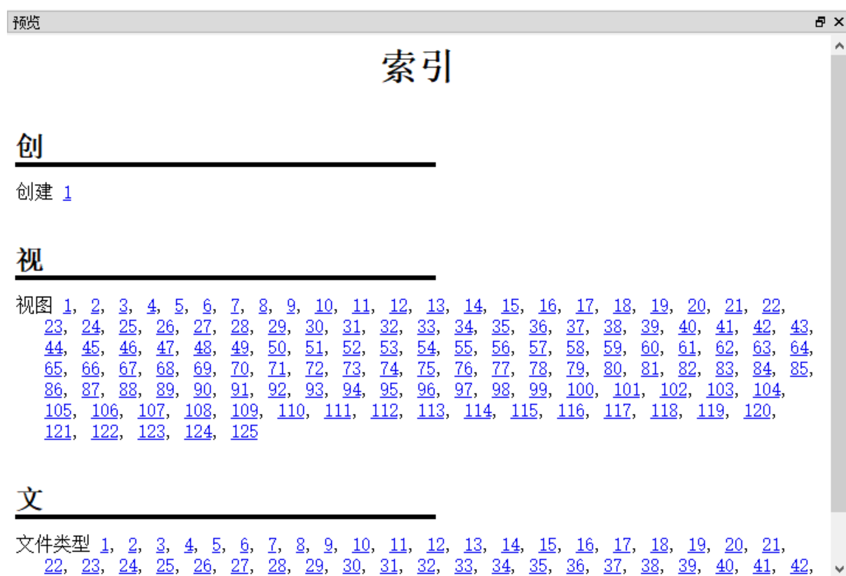


图5-75　「索引」内容文档

> 您可以像任何HTML页面一样编辑索引页面，但如果重新创建索引，您的更改将被覆盖为初始状态。

自定义索引样式表

索引页面使用单独的CSS文件（sgc-index.css）来格式化索引页面。如果文件已经存在于您的书中，则不会被覆盖（因此您可以自定义该文件，而不会覆盖更改）。但是如果不存在，它将在您的首选项文件夹中查找*sgc-index.css*，并将其用作默认样式文件。如果您的首选项目录中没有文件，则Sigil将为索引创建一个默认样式表。

索引编辑器

索引编辑器用于管理在索引中的条目列表。它不包括您标记为索引的单词——它们会自动包含在内。

要打开索引编辑器，请选择菜单项「工具→索引→索引编辑器」或其键盘快捷键方式打开：

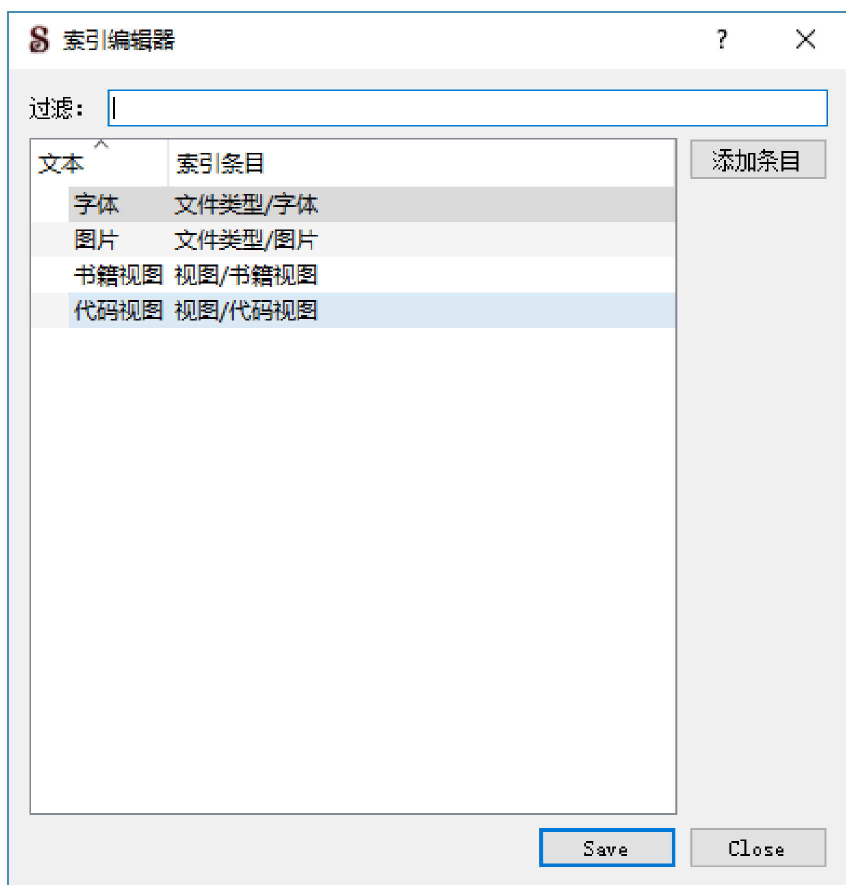


图5-76　索引编辑器

文本

「文本」列是您希望Sigil在文本中搜索的索引列表，然后在「索引条目」列中创建使用该格式的索引条目。「文本」列中的条目可以使用以下格式：

- 单词：书中的单词必须与列出的单词完全匹配（区分大小写）。不要包含任何HTML标签，只显示在「书籍视图」中的单词。
- 正则表达式：您可以输入大部分正则表达式——它们将逐段匹配。例如，为了在忽略区分大小写的情况下匹配单词，您可以使用「[Gg]utenberg」或「(?i)Gutenberg」。

> 您可以将文本粘贴到「文本」中。从包含单词列表的文件中，文本将自动分割为每行的单独条目。
> 您也可以使用「打开」打开索引文本（以及按Tab键分隔条目）。

索引条目

「索引条目」列是索引中实际显示匹配「文本包含」列的字符。

「索引条目」列中的条目可以使用以下格式：

- 空白：如果您将该条目留空，则「文本」列中输入的字将用于该条目的名称。
- 单词：如果您输入单词或短语，则将使用该单词或短语（例如Oak而不是oak）。
- 标题/词语：如果您在名称中使用「/」，则将创建一个分层组（例如，Forest/Tree/Oak将创建一个在Forest下Tree的条目Oak）。
- 标题/：如果在条目名称末尾使用「/」，则将创建一个分层组，「文本」列中的单词将用作子条目。

右键菜单

关于编辑索引的大多数操作都可以通过使用右键菜单来完成。

选择一个或多个条目，然后右键单击选择以下选项：

- 添加条目：在您单击的条目下添加空白索引。
- 编辑：编辑条目。
- 剪切：删除一个或多个稍后粘贴的条目。
- 复制：复制一个或多个稍后粘贴的条目。
- 粘贴：粘贴从剪切/复制的任何条目。
- 删除：删除条目。
- 打开：使用文件（保存的索引文件或文本文件）中的条目替换当前索引列表。
- 重新加载：恢复最后保存的索引列表。
- 另存为：将所有条目保存到文件中，以保存索引条目的列表。

- 自动填充：将列表中的每个单词填写在书中。
- 选择全部：选择所有条目（易于删除）。

保存

要将条目保存到计算机中，请单击Save。

要退出而不保存您的条目，您只需关闭窗口（使用Close按钮或Esc键）。您还可以使用右键菜单「重新加载」从计算机中重新加载条目——这将覆盖当前条目。

> 如果您保存并运行Sigil的多个副本，则Sigil的其他实例将自动重新加载新保存的索引条目，以使所有副本保持同步。

过滤

您可以使用「过滤」文本框来限制显示的索引列表，以便更容易找到索引。

打开或导入文本文件

右键菜单项「打开」可用于打开保存的索引文件，但也可以打开文本文件。

您可以列出文本文本中单词的文本——每行一个。或者您可以通过用Tab空格将它们分开来，一行包含「文本」和「索引条目」两个值。文本文件必须是UTF编码，不然打开后是乱码。

5.3.23　报告：查看有关您书的详细信息

Sigil提供了几个报告，其中列出了书籍中的文件和样式的详细信息。

通常您以双击报告中的条目以跳转到列出的文件名位置。

> 一些条目具有工具提示信息。例如在「图片文件」中，在条目列表中显示哪些文件使用图像。

使用「工具→报告」生成报告，然后选择要查看的报告：

全部文件

此报告列出了ePub中的所有文件、文件的大小、文件的位置和类型以及应用于文件的语义类型。

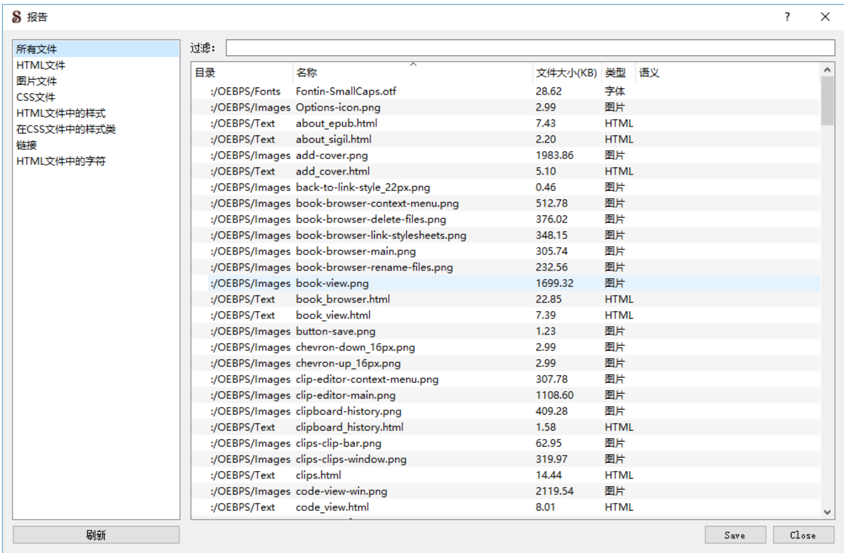


图5-77　「报告」窗口

HTML文件

此报告列出所有HTML文件和有关文件的详细信息。

报告

所有文件
HTML文件
图片文件
CSS文件
HTML文件中的样式
在CSS文件中的样式类
链接
HTML文件中的字符

过滤：

名称	文件大小(KB)	所有单词	拼错的单词	图片	视频	音频	样式表	格式完整的
titlepage.html	0.54	4	3	0	0	0	1	是
cover.html	0.58	1	0	1	0	0	0	是
cover_picture.html	0.66	1	0	1	0	0	1	是
rights.html	0.89	18	11	0	0	0	1	是
tutorials.html	1.47	146	17	0	0	0	1	是
clipboard_history.html	1.58	108	17	1	0	0	1	是
tutorial_add_cover.html	1.66	182	8	3	0	0	1	是
tutorial_generate_toc.html	1.84	165	4	4	0	0	1	是
tutorial_save.html	2.10	228	17	4	0	0	1	是
introduction.html	2.16	44	41	0	0	0	1	是
tutorial_validate.html	2.18	238	13	4	0	0	1	是
about_sigil.html	2.20	187	44	0	0	0	1	是
tutorial_mark_chapters.html	2.29	216	8	6	0	0	1	是
tutorial_getting_started_with_epub.html	2.46	377	33	0	0	0	1	是
tutorial_split.html	2.56	282	8	5	0	0	1	是
special_characters.html	2.87	173	27	4	0	0	1	是
installation.html	2.87	190	46	0	0	0	1	是
tutorial_metadata.html	2.92	278	15	5	0	0	1	是
tutorial_load_file.html	3.18	367	28	5	0	0	1	是
tutorial_tips.html	3.34	403	29	4	0	0	1	是
contributing_to_sigil.html	3.47	386	36	0	0	0	1	是
tutorial_spelling.html	3.57	414	18	3	0	0	1	是
tutorial_convert_to_html.html	3.68	529	26	0	0	0	1	是
external_editors.html	4.27	350	57	3	0	0	1	是
tutorial_find_replace.html	4.49	515	8	7	0	0	1	是
add_cover.html	5.10	437	111	1	0	0	1	是
tutorial_embed_fonts.html	5.58	692	42	2	0	0	1	是

刷新　Save　Close

图5-78　「HTML文件」报告

您可以从ePub的报告中删除文件，方法是右键单击所选文件，然后选择「从书中删除」。

图片文件

此报告列出您的所有图像，并显示它们是否在书中使用以及使用了多少次。您可以双击一个条目跳转到HTML文件中图像的第一次出现（不是CSS文件中的图像）位置。

报告

所有文件
HTML文件
图片文件
CSS文件
HTML文件中的样式
在CSS文件中的样式类
链接
HTML文件中的字符

过滤

名称	文件大小(KB)	使用次数	宽	高	像素	颜色	图片
insert-hyperlink_22px.png	0.36	5	22	22	484	Color	
back-to-link-style_22px.png	0.46	5	22	22	484	Color	
heading-1_22px.png	0.55	2	22	22	484	Color	h1
heading-normal_22px.png	0.55	1	22	22	484	Color	p
insert-id_22px.png	0.91	5	22	22	484	Color	
document-validate_22px.png	1.03	3	22	22	484	Color	
generate-toc_22px.png	1.04	3	22	22	484	Color	
button-save.png	1.23	5	29	29	841	Color	
Options-icon.png	2.99	2	27	27	729	Color	
chevron-down_16px.png	2.99	2	27	27	729	Color	
chevron-up_16px.png	2.99	2	27	27	729	Color	
note.png	4.04	1	197	124	24428	Grayscale	Text Note
tutorial-toc-empty.png	4.46	1	215	204	43860	Color	
thumbnail+-.png	4.69	2	53	22	1166	Color	
document-save_22px.png	4.80	3	35	34	1190	Color	

刷新　缩略图尺寸：　Save　Close

图5-79　「图片文件」报告

您可以从ePub的报告中删除文件，方法是右键单击所选文件，然后选择「从书中删除」。

CSS文件

此报告列出了所有的CSS样式表，并指出它们是否被链接以及使用的次数。

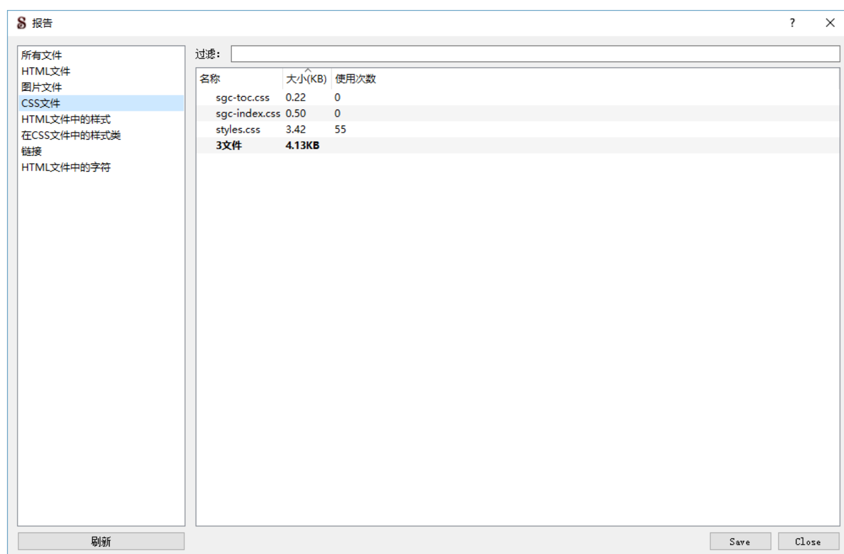


图5-80　「CSS文件」报告

您可以从ePub的报告中删除文件，方法是右键单击所选文件，然后选择「从书中删除」。

HTML文件中的样式

此报告扫描您书中的HTML文件，并提取文件中使用的样式名称列表。对于每个类名，它列出是否在CSS样式表中或在内联样式中找的到。

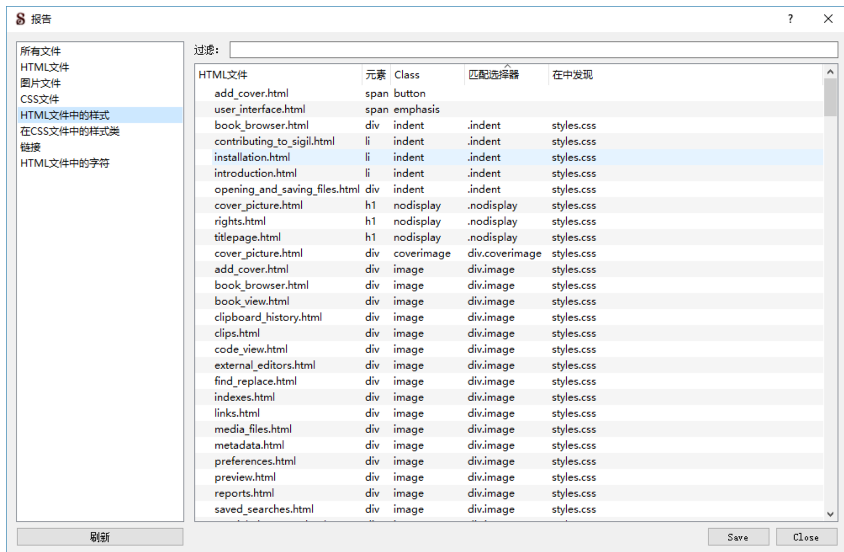


图5-81 「HTML文件中的样式」报告

在CSS文件中的样式类

此报告将扫描您书中的CSS样式表，并显示列出的样式是否在本书中使用。

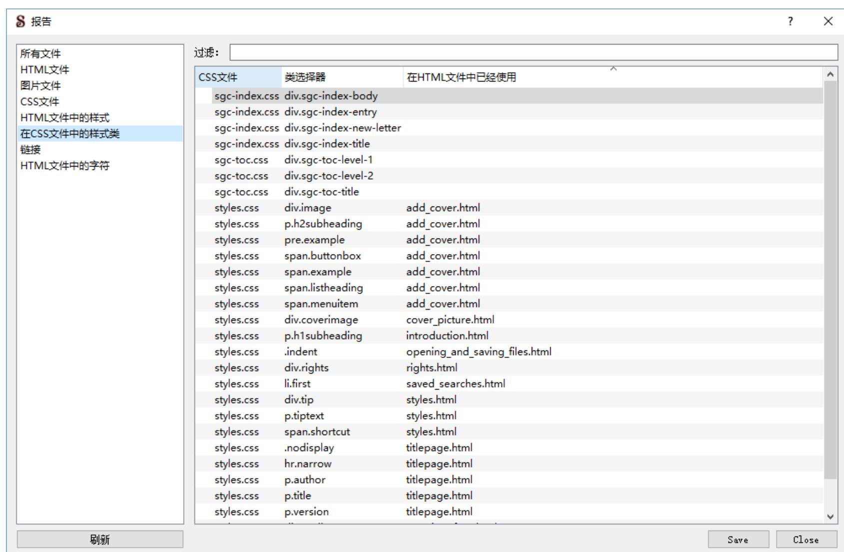


图5-82 「在CSS文件中的样式类」报告

您可以在报告中通过右键单击所选样式并选择「从样式表中删除」，从样式表中删除特定样式类。

链接和脚注

此报告列出在您的HTML文件中找到的每个链接（href）。这些代码中的「<a>」标签。它包括链接是否具有id，链接的目标是什么，以及该目标是否存在。

另外，为了帮助脚注或尾注，它检查反向链接。反向链接是指向原始链接目标的链接。换句话说，如果您为脚注创建了这样的链接，则报告可以确认它们是否彼此链接：

```
<!--链接文字：-->
<a id="t1" target="../Text/footfile1.xhtml#f1">text to footnote</a>
<!--链接脚注：-->
<a id="f1" target="../Text/file1.xhtml#t1">footnote back to text</a>
```

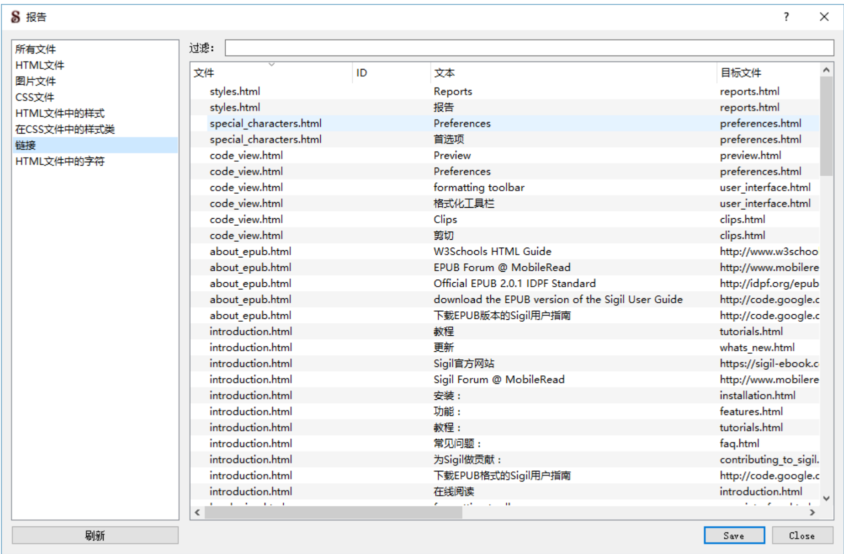


图5-83 「链接」报告

HTML文件中的字符

此报告提取HTML文件中使用的所有字符列表。更具体地说，它列出了在「书籍视图」中显示的任何字符。如果HTML实体代码存在，则列出它们的十进制和十六进制代码。所有字符都列在底部，便于剪切和粘贴。

此报告可用于识别书中是否正在使用额外的字符，如果您正在子集字体，则可能会有所帮助。

图5-85　「HTML文件中的字符」报告

5.3.24 外部编辑：使用「打开方式」编辑文件

Sigil虽然是一个强大的编辑器，但它不能编辑图像，它可能没有您在其他编辑器中使用的一些功能。在这些情况下，您可以在外部程序中编辑文件——保存时，Sigil将立即更新其副本。如果您想要快速编辑图像或想编辑CSS样式表，并立即看到修改如何影响图书的布局，这非常有用。

要在外部程序中编辑文件，请在「图书浏览器」中右键单击文件的名称，然后从右键菜单中选择「打开方式」（您还可以在「书籍视图」中右键单击图像）：

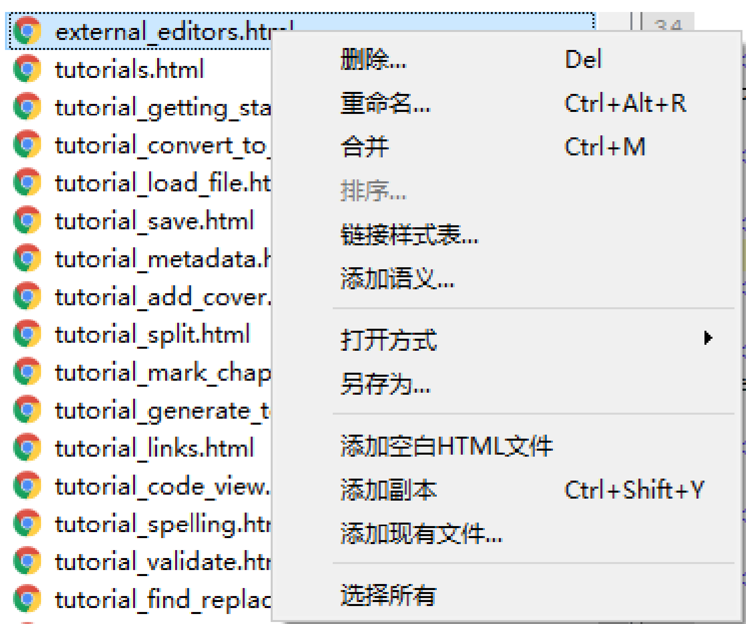


图5-86 内容文档右键菜单

当您第一次打开文件时，首先需要选择编辑或浏览该文件类型的应用程序。每个文件类型（html、image、stylesheet）都可以使用不同的应用程序。

然后，您可以给应用程序设置一个名称，将在下次使用「打开方式」时出现在菜单项中：

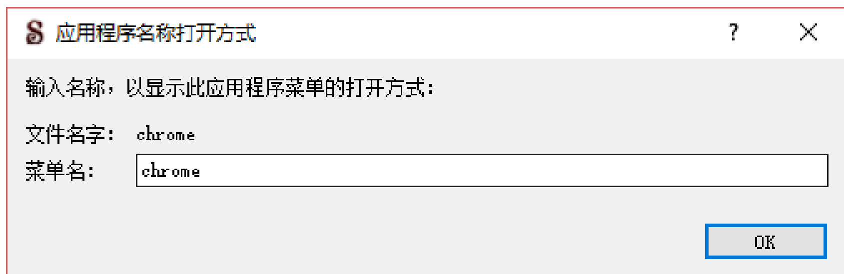


图5-87　内容文档打开方式

当下次在文件中使用「打开方式」时，您将看到您设置的应用程序名称，以及选择其他应用程序的选项。这将只列出一个默认应用程序，但您可以随时更改：

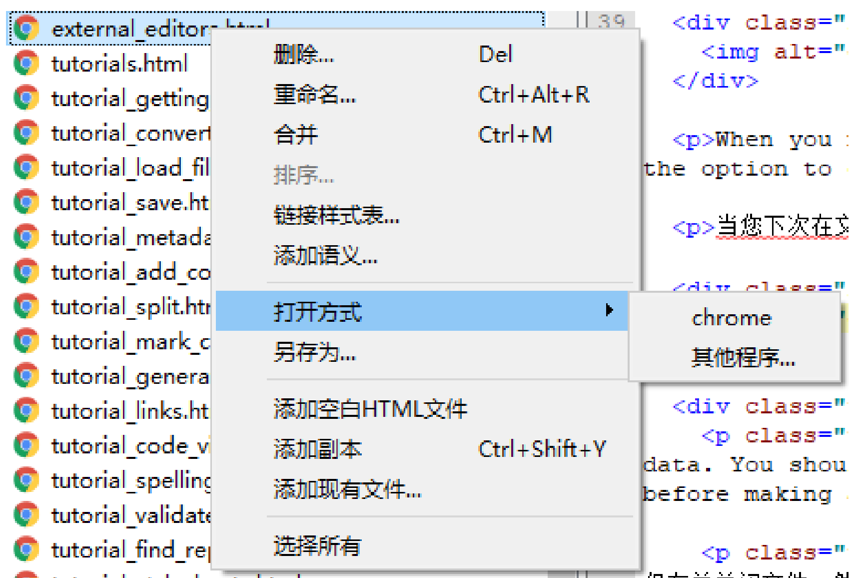


图5-88　选择内容文档打开方式

如果您在另一个程序中编辑文件时对Sigil中的文件进行更改，那么可能会丢失数据。您应该始终用「打开方式」进行外部编辑更改，然后在外部应用程序中保存并关闭文件，然后在Sigil中再对该文件进行更改。

在某些情况下，图书可能无法在「书籍视图」中及时更新，尤其是在「预览」打开时。您可能需要先切换到「代码视图」然后返回「书籍视图」或关闭当前选项卡并重新打开文件以查看编辑过的图像。

5.4 教程——如何利用该指南

创建您的第一个ePub电子书，并学习使用Sigil与本节中的教程。

有关Sigil所有功能的详细信息，请参见功能一章。

为了让您了解制作ePub所涉及的内容，以下概述是您可能要遵循的步骤：

- 将您的文档转换为HTML（如果它不是ePub），那么可以将其载入Sigil。
- 填写书籍的作者、标题等详细信息。
- 添加封面图片
- 创建目录
- 根据ePub标准验证您的电子书。
- 在Sigil中预览和测试效果。
- 在实际的电子阅读器上进行兼容性测试。

> 看看ePub是什么样子，试试Sigil，您可以下载ePub版本的Sigil用户指南，并在Sigil中打开它。

5.4.1 ePub入门

如果您要创建ePub电子书，了解ePub是什么以及Sigil可以帮助的方式可能是个好主意。

ePub是用于显示书籍的标准文件格式，Sigil可帮助您创建ePub。Sigil旨在将您在文字处理器中创建的书籍转换为标准的ePub文件，该文件可以在电子书阅读器上发布和阅读。它也可用于清理和重新格式化现有的ePub。

这取决于您需要「清理」ePub的需求。如果您只是为自己排版一本书，您可能不会关心太多的细节，使用Sigil就足够让它在您的电子阅读器上显示了。但是如果您要格式化出版物，那么您会发现Sigil可以轻松访问和修改有关您书籍的任何细节。

当您在典型的文字处理程序中编辑文档时，通常会出现1个文件，例如`mybook.doc`。此文件包含有关您的文档的所有信息。ePub略有不同，它不是仅使用1个文件来存储您的信息，它使用单独的文件来存储构成文档的部分：章节、图片、目录等。但是，由于与人共享大量单独的文件是很困难的，所有这些单独的文件实际上存储在单个压缩zip文件中，文件扩展名为`.epub`，例如`mybook.epub`。一个ePub文件实际上只是一个zip文件，其中包含一本书所需的所有文件。

Sigil可以打开并创建ePub文件，它允许您编辑、重新排列和格式化ePub中的各个文件（如文本和目录）。

将文档复制或导入Sigil后，它将成为ePub文件的一部分。当您将书保存在Sigil中时，您的文档将保存在ePub文件中。

请注意，Sigil旨在与ePub 2（大多数电子墨水阅读器支持的版本）兼容，并可轻松转换为其他格式。它还支持一些ePub 3功能，如音频和视频——但这些可能在大多数设备上不受支持。

5.4.2 准备在Sigil中打开的文件

如果您已经有ePub或HTML文件，或者就是从Sigil开始编辑文档，您可以跳过此步骤。

如果您开始使用的是文字处理器创建的文档，那么首先需要做的是将文档转换为HTML，以便将其导入到Sigil中 一些程序可以生成ePub文件。一些程序有附加组件或功能来清理并输出HTML文件。输出程序决定了HTML或ePub（或甚至纯文本）的质量——您可以随时测试不同的方法来查看最有效的方法。

> 为了更容易地转换为ePub，您的HTML文件应该「更干净」，并且在多个设备上尽可能正确显示。这意味着使用样式进行格式化，而不是使用太多花哨的格式和布局。
> 不要指望ePub看起来像打印的书一样——ePub的设计使读者能够控制书的显示方式。

虽然转换不是Sigil的一部分，但是有一些方案可以转换导入Sigil的文件。

Word文档

使用微软的Word菜单项「文件→另存为（筛选过的网页）」。或使用MS Word Macro @ MobileRead。

这两种方法都尝试减少由微软Word导出的额外代码的数量，以便您的HTML文件在Sigil中可以更清洁和更简单的编辑。你真的想让你的代码尽可能干净，因为Word添加了许多额外的HTML标签，你可能会在某些时候删除掉。

当使用Word或任何文字处理器时，您应该始终尝试使用样式来格式化代码，而不是仅仅使一些文本加粗、更大一些等等。格式化ePub时，样式也是一个重要的概念。

文本文件

并不是必须要把文本文件转换为HTML。Sigil可以直接打开「.txt」文件。

> 文本文件中必须在段落之间具有空白行，以便Sigil可以将它们分割成单独的段落。文本文件编码必须是Unicode编码。

文本文件不是理想的格式，但它们仍然可以比某些格式要好，因为它们没有HTML标签。如果其他格式有许多问题，文本文件可能值得使用。

有时候，为干净的文档添加样式比尝试清理转换的文件更快。

其他格式

许多其他文字处理器（如Libre Office）也可以保存或输出HTML，甚至有保存为ePub的功能，您可以尝试下。您还可以使用其他软件（如calibre）将其他格式转换为HTML或ePub。

请注意，任何自动转换的文件将会引入大量额外的代码。如果您只是格式化一本书供个人使用，您可能不会关心此事。但是，如果您为了专业目的而格式化书籍，或者希望有尽可能少的额外代码，那么应该避免这种情况。

> Sigil只支持导入HTML文件，因为从其他格式转换实际上是非常专业的，最好留给其他工具。

5.4.3　用Sigil打开你的文件

一旦你有一个HTML或ePub文件，是时候把它加载到Sigil了。即使您已经有ePub文件，您应该按照以下快速介绍的步骤使用Sigil。

> 如果您正在使用ePub文件，请在Sigil中打开前进行文件备份。

启动Sigil

当您第一次启动Sigil时，将显示一个带有多个窗口的屏幕：

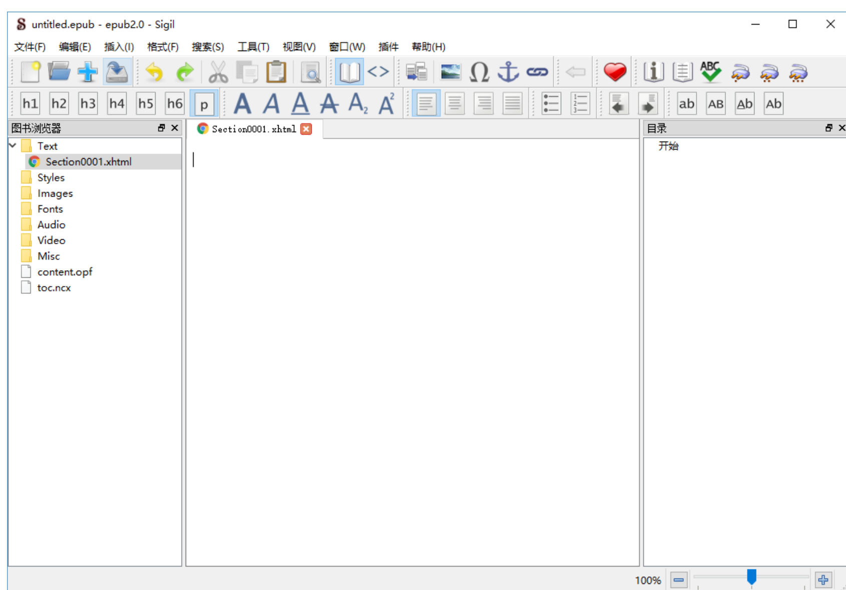


图5-89　Sigil初始界面

左侧的「图书浏览器」窗口显示您ePub中的所有文件，编辑窗口位于中心，并且默认为「书籍视图」（以编辑您的图书），因为您尚未加载任何文件，它显示为空白，右侧的是目录窗口（以显示您定义的章节标题）。

如果打开了ePub，Sigil会如下显示：

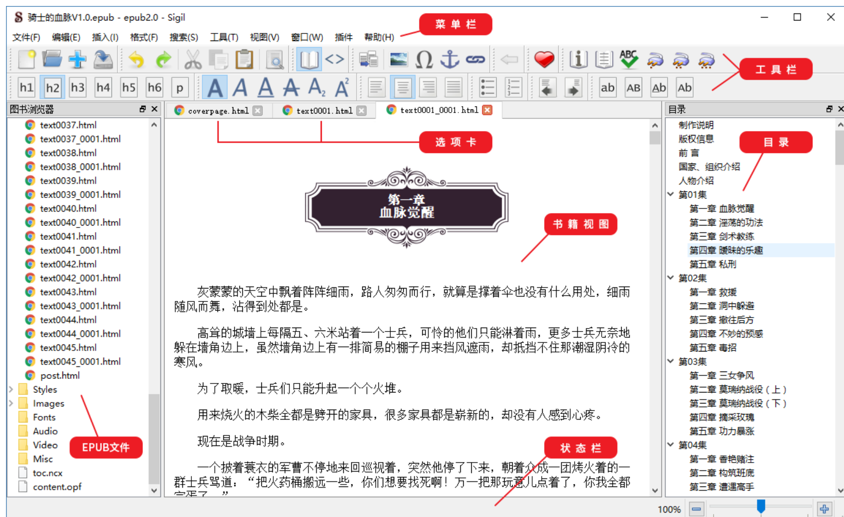


图5-90　Sigil各窗口名称

您可以通过在「图书浏览器」中双击它们的名字，打开书中的各个文件以进行编辑。

> 有关详细信息，请参阅用户界面章节。

加载你的文件

要将您的HTML或ePub文件加载到Sigil中，请单击打开按钮或使用菜单项「文件→打开」：

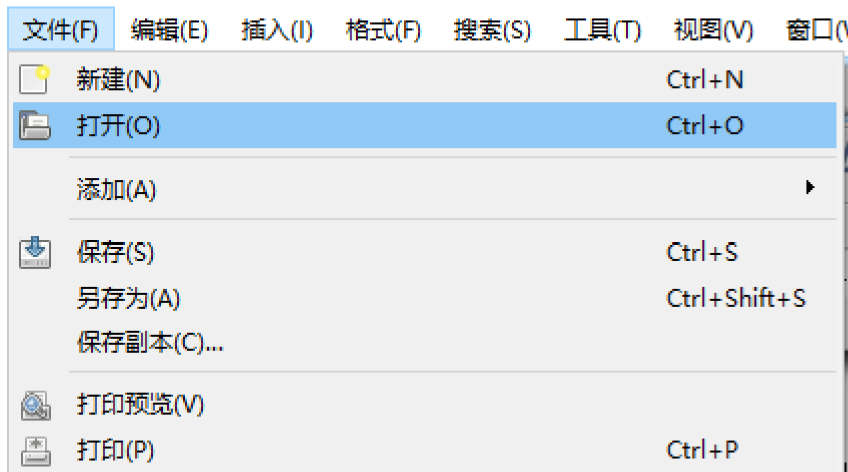


图5-91　「文件」菜单选项

从文件管理器中选择要打开的文件，例如*TestBook.xhtml*。

现在，您将在左侧「图书浏览器」窗口中的Text文件夹下看到您的HTML文件。该文件将在中间的「书籍视图」编辑器中打开，文件名显示在标题栏上：

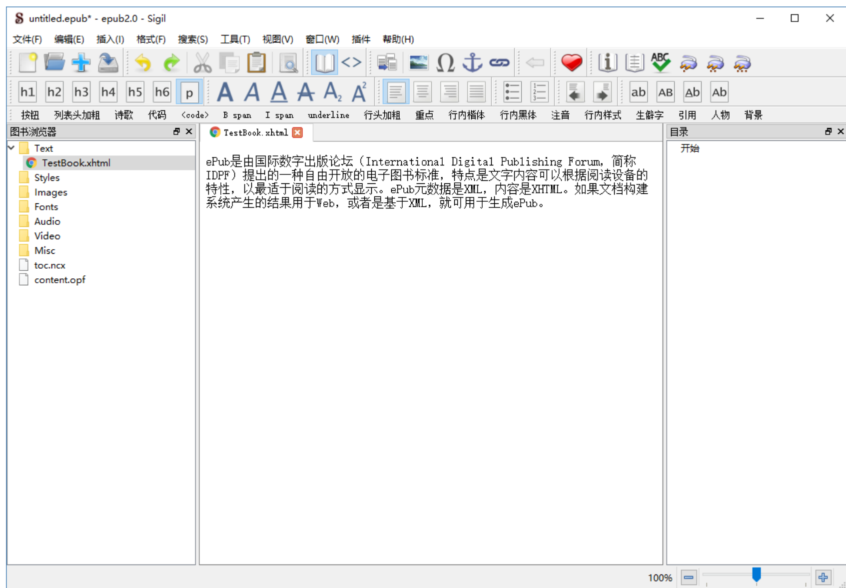


图5-92　书籍视图

您的HTML文件现在已经是ePub的一部分了。从现在开始，您正在编辑您的ePub，而不仅仅是一个HTML文件。您可能已经注意到，当您在Sigil中打开文件时，它创建了ePub格式所需的其他几个文件，您可以在「图书浏览器」中看到原始文档旁边的其他文件。

> 如果您的文件包含任何图像或样式表等，Sigil将尝试加载它们，并将其存储在「图书浏览器」中相应的文件夹中。
> 将文件转换为ePub或加载ePub时，Sigil可能会添加一些必要的详细信息，并将文件组织到标准文件夹名称中。

5.4.4 保存您的ePub文件

现在你已经把你的文件加载到Sigil了，你应该先保存它。

> 加载新文件后，您应该始终执行保存——即使是刚开始编辑ePub，这会确保保存时有没有问题发生。

由于您正在编辑的文件是ePub（即使您只加载了一个HTML文件），当您在Sigil中保存时，您将保存ePub格式的文件（尽管可能会丢失一些您需要添加的细节）。

要保存您的ePub，请单击保存按钮或使用菜单项「文件→保存」：

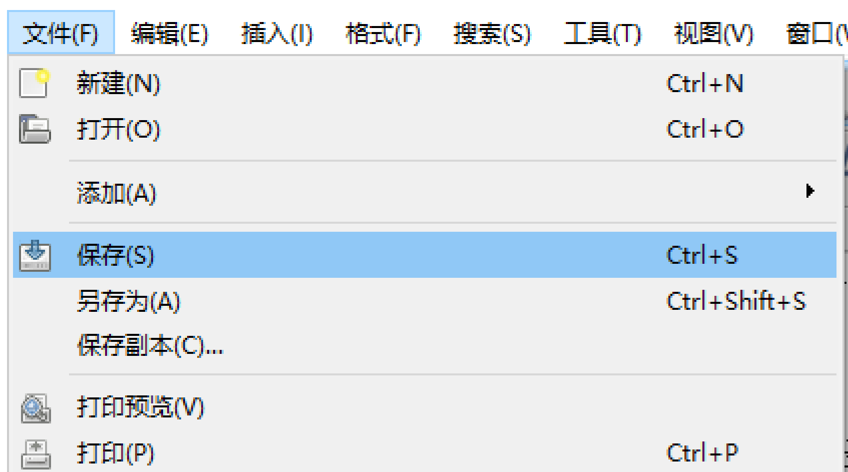


图5-93　「文件」菜单选项

如果您加载了一个HTML文件，您的ePub也不会有一个名称——因此您将被提示输入文件的名称。它将被赋予一个.epub扩展名，因为它将被保存为一个ePub文件。

> 您可以随时单击保存按钮来保存您的ePub——您应该经常保存。

请注意，您的图书标题（位于窗口顶部）已更新（例如`TestBook.epub`），状态栏会简要显示「EPUB已保存」：

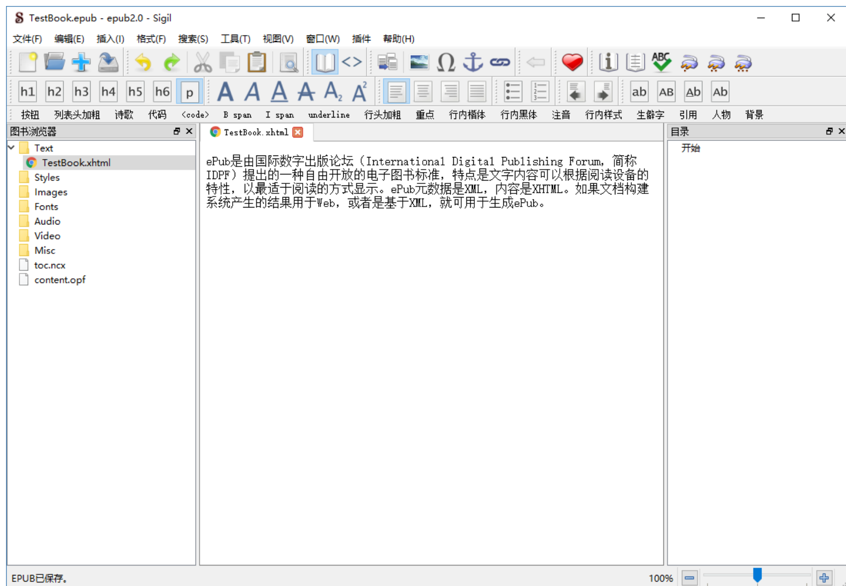


图5-94　保存后的epub文件

你现在有一个ePub文件——但不要停下来！

为确保您的电子书符合ePub标准，您需要添加更多信息，如以下教程所示。

5.4.5 添加作者和标题

除了书的实际内容外，ePub文件还包含书籍详细信息，如作者和标题。这些可以使用Sigil来添加到ePub中。

要将关于图书的详细信息（称为元数据）添加到您的ePub中，请单击「元数据编辑器」按钮或菜单项「工具→元数据编辑器」



图5-95　「工具」菜单选项

将显示「元数据编辑器」窗口：

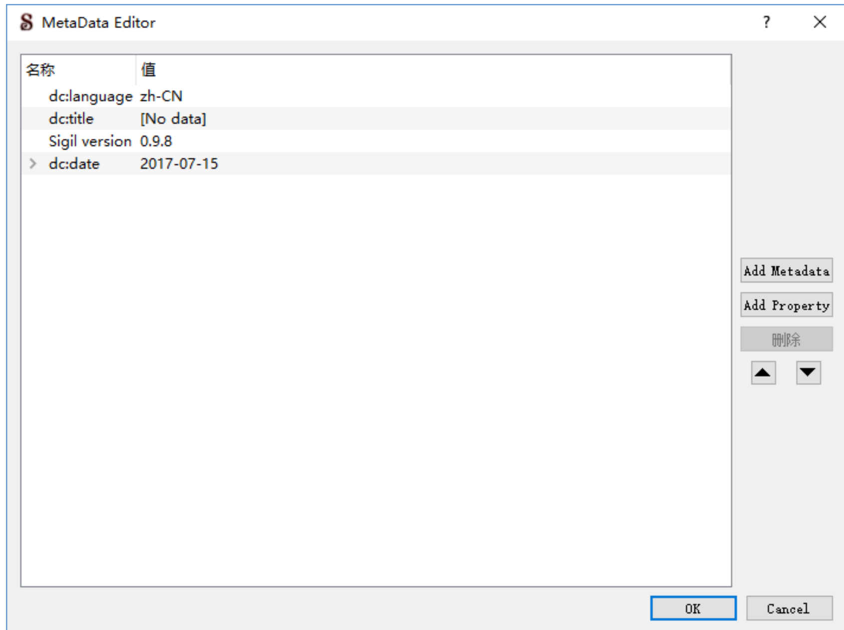


图5-96　元数据编辑器

您可能会注意到，「元数据编辑器」已经有一些关于书的细节——上次修改该文件的日期。每次在Sigil保存书籍时，都会添加或更新。您可以添加许多其他的详细信息，但ePub所需的条目是标题、作者和语言。

要添加标题、作者和语言到您的书中：

- 标题：输入书的完整标题。
- 作者：输入作者姓名，例如输入Lewis Carroll。
- 语言：从下拉框中选择书籍的主要语言。
- **File-As**：这实际上是可选的，所以它可以留空。但是通常包括它，因为它控制作者的名字是如何排序的。书入作者的名称，因为它应该被排序，例如输入Carroll,Lewis。

这是您在填写数据时的界面：

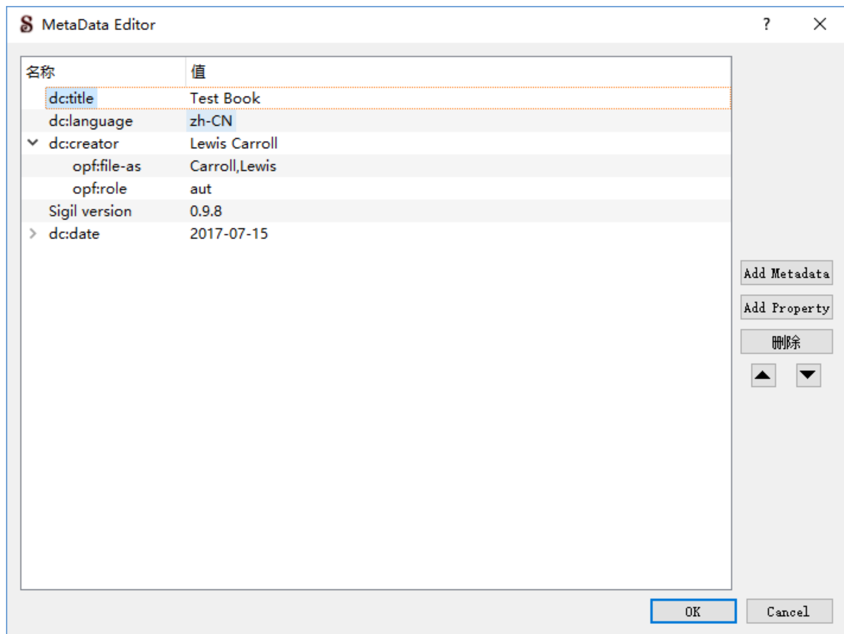


图5-97　填写图书元数据

更新信息后，您可以单击OK，然后是保存按钮——再次保存您的ePub。

> 您可能已经注意到，元数据编辑器中还有其他值。ePub只需要标题、作者和语言，但是如果需要的话，点击Add Metadata或Add Property按钮添加更多的值。
> 有关详细信息，请参阅元数据编辑器一章。

5.4.6　添加封面图片

每本书都应该有一张封面图片，而Sigil可以轻松为ePub添加一个封面。您需要有一个封面图片可供使用（并有权使用它）。封面的确切尺寸准则取决于您和设备。

要添加封面，只需选择菜单项「工具→添加封面」：

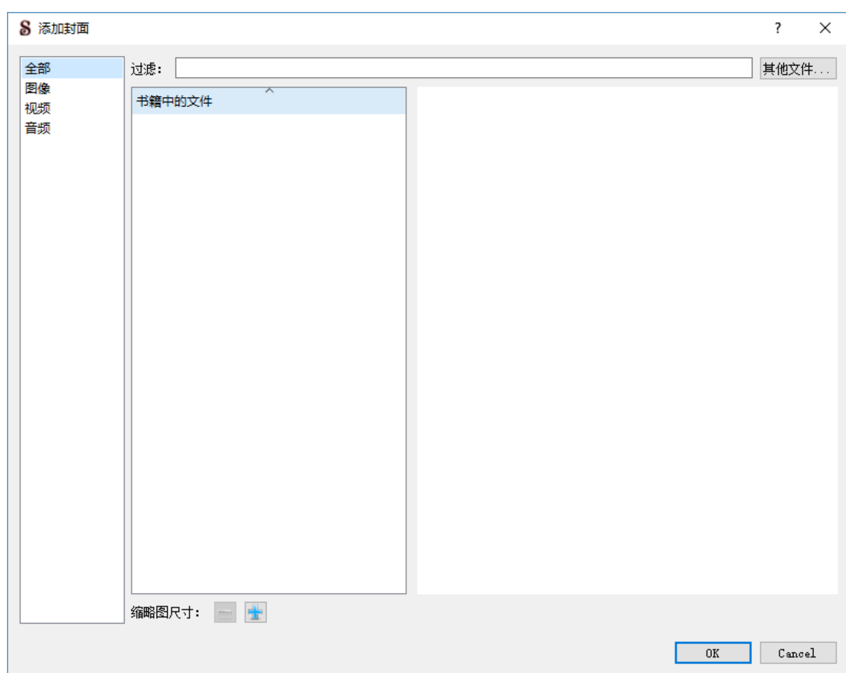


图5-98　「添加封面」窗口

「添加封面」窗口会显示ePub中已有的图像列表。但是由于你还没有加载任何图像，它是空的。

您可以将图像插入到ePub中，并通过单击其他文件按钮一步步地覆盖图像。

从文件管理器中选择封面图片，然后单击OK。

如下，Sigil已将您的图像添加到ePub中并将其标记为封面图片。它还在您书开头位置创建了一个HTML文件，并将您的封面图像链接到文件中，以便在打开本书在第一页时显示：

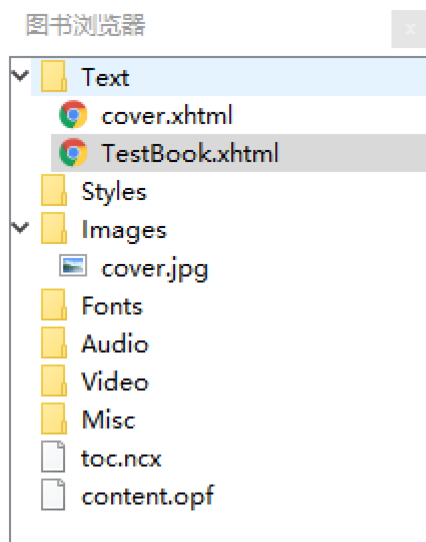


图5-99　「图书浏览器」窗口

你应该点击保存按钮保存你的书。

5.4.7　为每一章创建单独的文件

如果您将一个HTML文件加载到Sigil中，这通常意味着您的所有文本和章节都在一个大文件中。虽然这是可以的，但是ePub的标准做法是将书中的每一章分成一个单独的HTML文件。一些电子阅读器需要较小的文件来更容易地显示阅读您的书籍，同时编辑也更为方便。

> 由于大多数电子阅读器在新页面上加载新文件，所以这意味着您的章节也将在新页面上开始。

每章一个文件

将书拆分成单独文件的最简单的方法是使用「从光标处分割」按钮。

例如，要在新页面上显示章节，请单击章节标题的左侧，以便您的光标是位于行的开头：

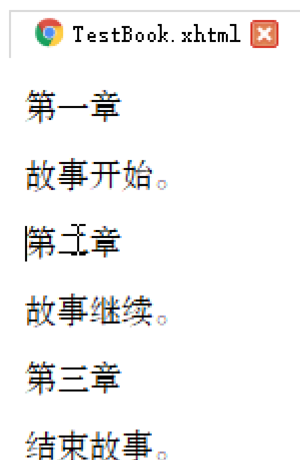


图5-100　拆入光标在拆分处

现在点击「从光标处分割」按钮，将文本分为两部分：

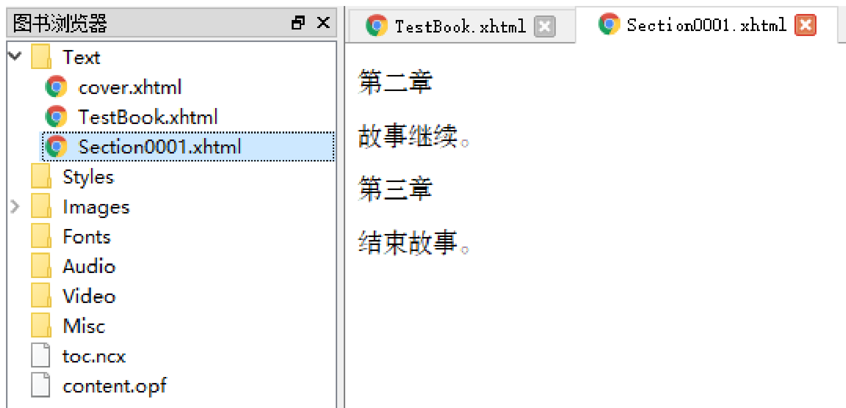


图5-101　执行拆分文件操作

你可以看到，拆分之后的所有内容现在都在一个名为「Section0001.xhtml」的新文件中。

如果你看原始文件，你可以看到它只包含分割标记之前的文字：

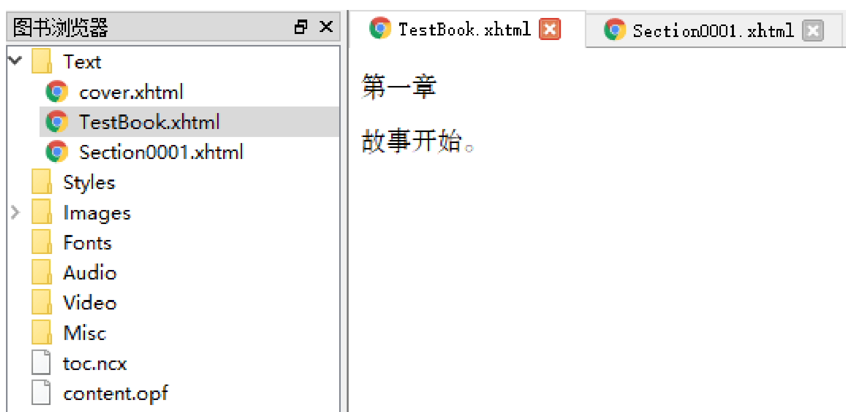


图5-102　光标前的内容成为一个单独内容文档

您可以继续拆分书中的每一章，并将任何应该需要从新页面开始的部分放在单独的文件中。

如果要重命名文件，请参阅重命名部分。这包括如何一次重命名多个文件，例如Chapter_1、Chapter_2等。

> 有关详细信息，请参阅拆分与合并一章，包括如何插入拆分标记并一次拆分每个文件。

5.4.8 识别你的章节

ePub可以提供目录（简写为TOC），以帮助读者在书中的章节之间跳转。Sigil可以从您的电子书中自动显示和生成目录——您只需要设置好章节标题。

Sigil在单独的窗口中显示目录——右侧为默认情况：

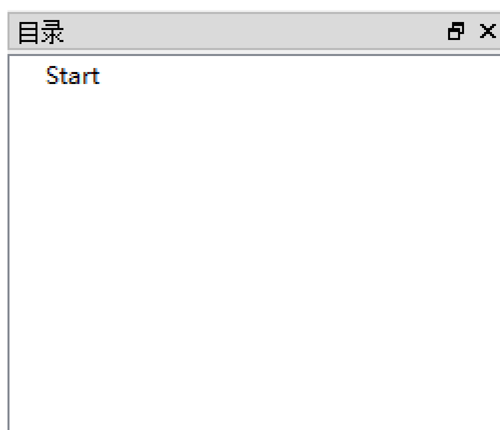


图5-103　「目录」默认窗口

标记你的章节

为了创建一个目录，你需要告诉Sigil你的章节名称是什么。要做到这一点，只需在书中选择每个章节名称，然后使用其中一个工具栏按钮将其标记为章节标题。

只需点击想要设置为章节标题所在段落的任意位置：

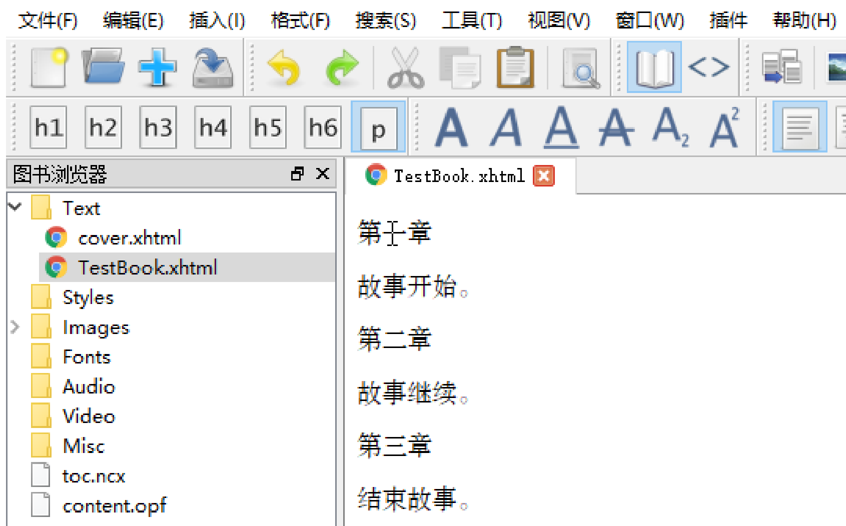


图5-104　格式化标题

点击「一级标题」按钮 h1 将文本标记为章节标题：

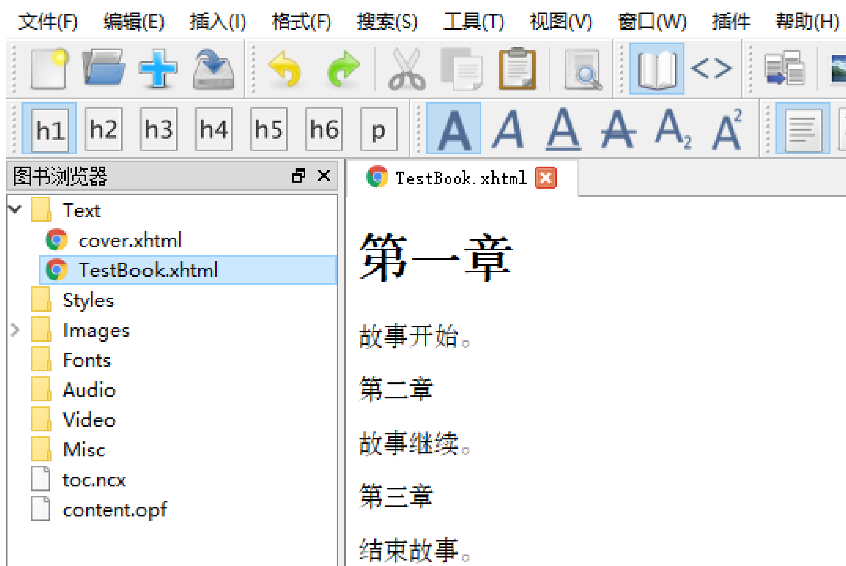


图5-105　使用标题格式化按钮

然后，您可以在书中标记剩余的章节标题：

TestBook.xhtml

第一章

故事开始。

第二章

故事继续。

第三章

结束故事。

图5-106　格式化标题

> 当您点击文本中的标题时，标题按钮将突出显示标题级别。

其他标题级别（h2、h3等）允许您创建子章。段落按钮 p 可用于将文本更改为正常段落而不是标题。

> 如果您的书中有多个HTML文件，只需在「图书浏览器」中双击它们即可打开。

5.4.9　生成目录

为了让Sigil可以生成目录，您必须已经确定了章节名称。如果您还没有这样做，请参阅上一个教程。

标记所有章节标题后，您可以单击「生成目录」按钮或使用菜单项「工具→目录→生成目录」以打开新对话框：

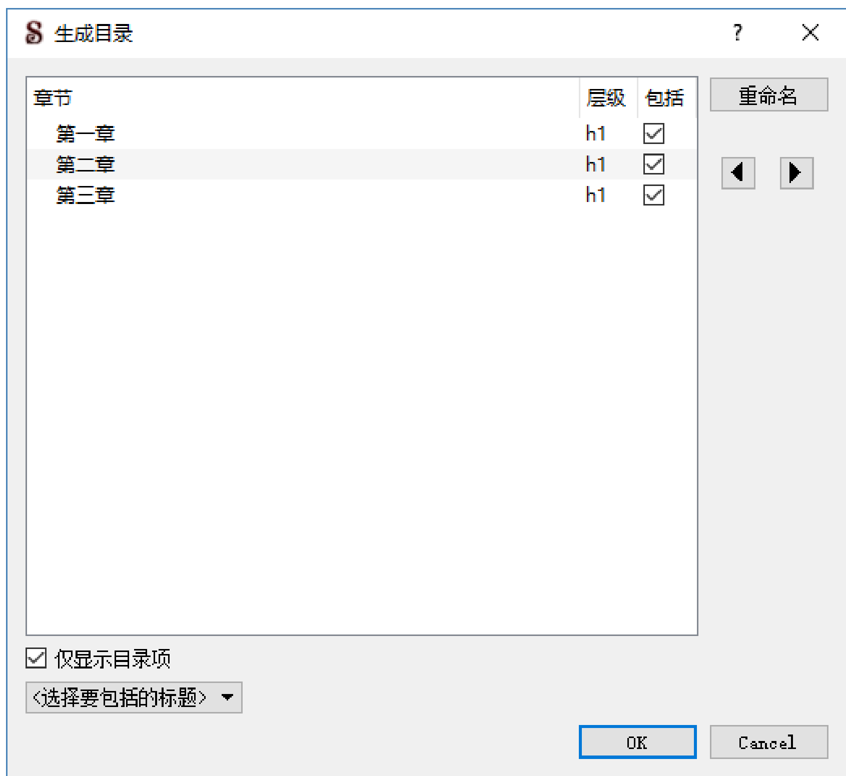


图5-107　生成目录

该对话框允许您选择需要包含或不包含在「目录」中的标题。

完成后单击OK。

目录窗口将被更新，以显示您当前的目录——其中包括您选择的标题层级：

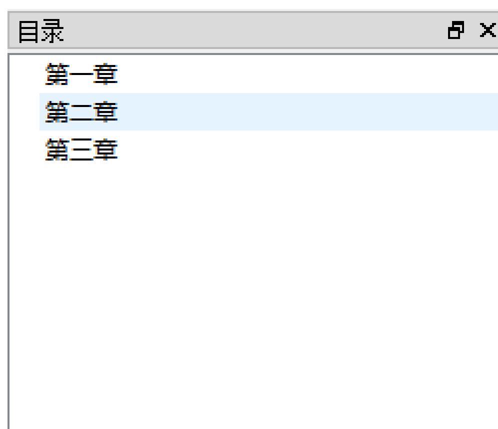


图5-108　「目录」窗口

如果您点击「目录」窗口中的条目，您将跳转到标题所在书中的位置。每当您更改标题时，都应重新生成目录。

点击保存按钮——如果您还没有保存书籍。

> 有关详细信息，请参阅目录一章。

5.4.10 创建链接和注释：链接您的文本

Sigil允许您轻松地将文本链接到书籍的其他部分，例如章节名称和注释，以便读者可以点击链接并跳转到相应的文本位置。

ID

为了让Sigil创建一本从你书中的某些单词到你书中其他地方的链接，你首先需要给你想要链接的文本添加一个名字或ID。

如果您链接的目标只是文件的顶部，那么您不需要执行任何操作，因为您可以使用文件名作为目标ID。

但是，如果您想指向书中的特定单词，则需要为文本分配名称或ID。

> ID只是一个独特的名称，如note1，您将分配给您的一些文本，以便您可以参考它。ID名称在HTML文件中必须是唯一的（并且以字母开头），但是如果您移动文本，则它们最好在整个ePub中也是独一无二的。
> ID和链接通常使用HTML锚标签（<a>）创建——因此ID通常被认为是链接的锚点ID，因此用于插入ID的按钮。

创建一个ID

要创建ID，请选择书中的一些文字，然后点击插入ID按钮⚓：

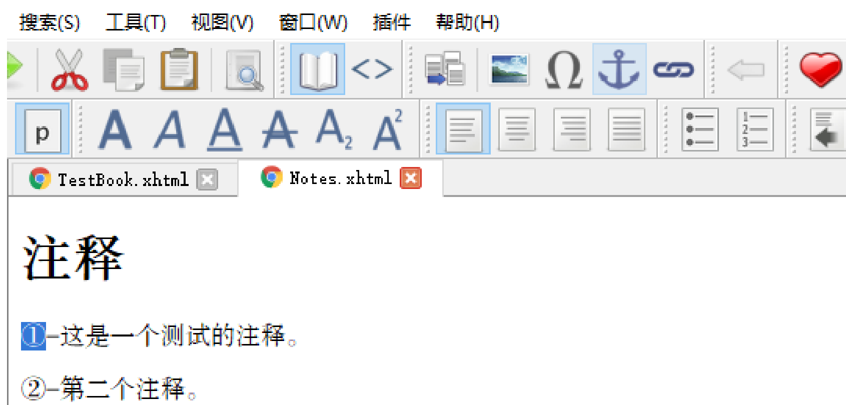


图5-109 创建注释ID

这将打开「插入ID」对话框，您可以在其中键入ID的名称：

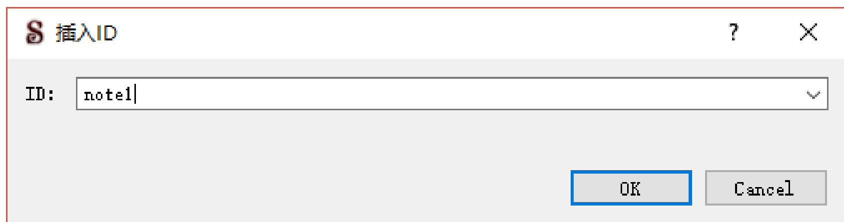


图5-110　插入ID

您在「书籍视图」中看不到任何文字上的更改，但如果您单击文本并再次单击「插入ID」按钮，您将看到列出的ID。

> 为了在「书籍视图」中删除锚标签及其ID，您需要选择所有文本并将其删除并重新输入。或在「代码视图」中直接删除HTML代码。

链接

链接是指向您的ID的一种方式，当读者点击时，它会「跳转」到ID的位置。

要在Sigil中创建一个链接，请选择一些用于链接的文本，然后单击「插入链接」按钮：

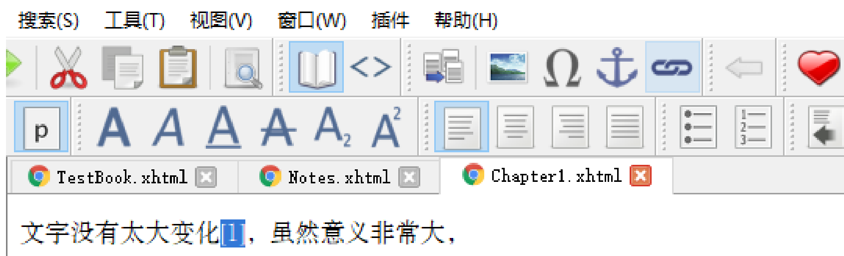


图5-111　文中注释链接插入

这将打开「选择目标」对话框，其中会显示可以链接的ID列表：

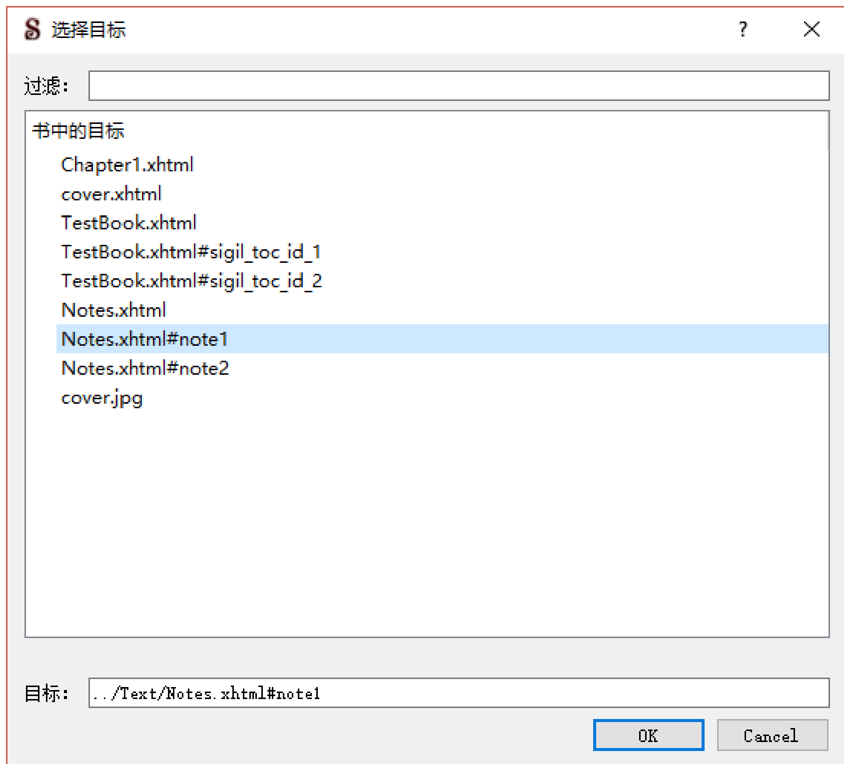


图5-112　选择注释链接目标

您会在列表中注意到一些不同的ID。起初它可能看起来不像你看到的那样，所以找到它，它有助于了解一个链接的样子：

- **Filename.xhtml**：它指向一个文件名作为链接的目标，并指示链接跳转到文件的顶部。这是使用「生成目录」出现的链接。
- **Filename.xhtml#ID1**：这指向文件中的特定ID。当您使用此链接作为链接的目标时，阅读器会将文本转到列出文件中ID的确切位置。
- **#ID1**：这只是ID名称，并且意味着它与链接在同一个文件中，因此不需要文件名。

当您选择链接时，您将看到「目标」文本框显示将要使用的实际链接。单击OK，或双击一个条目，将为您的文本创建链接。

如果您点击文本中新创建的链接，您将跳转到ID的位置。然后，您可以点击返回到您点击链接的位置。

反向链接

大多数电子阅读器都有一个「后退」按钮，您可以在按一下链接回到原始的链接文本。但是对于那些没有设置的书来说，您可以创建一个从您的目的地回到原始源文本的链接。这通常被称为反向链接。

反向链接就像任何其他链接——这意味着您必须创建从源文本到目标注

释的链接，还可以从注释到源文本的链接，因此源和目标都将需要ID和链接，像彼此的镜像版本：

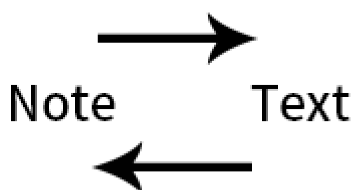


图5-113　跳转示意图

例如，在创建典型的尾注情况下：

1. 在您的书籍（或章节）的最后选择您的注释，并为其分配ID：note1。
2. 在书中选择您的文字，并为其分配ID：text1。
3. 再次选择相同的文字，并创建一个指向该注释ID的链接。
4. 再次选择您的注释（点击链接跳转到注释并测试链接），并创建一个指向您的文本ID的链接。
5. 点击注释，来测试让你回到文本的链接。

5.4.11 快速介绍代码视图

到目前为止，您可能已经正在浏览您的文字，正如电子书阅读器上显示的书籍——在Sigil在使用所见即所得「书籍视图」模式。

但是，您的文本实际上并不存储在ePub中，就像在「书籍视图」中一样。实际上，它是以HTML格式存储的——这是一个混合了文本和HTML标签的，告诉阅读器如何解读和显示你的书。

您可以使用「代码视图」查看文件中的HTML代码。

那么为什么你需要知道这一点呢？因为为了方便使用「查找&替换」或控制书籍的样式，甚至理解「拼写检查」，您需要熟悉「代码视图」。

> 关于「代码视图」值得您去学习一下，以帮助您了解格式化的工作方式。如果您想要使用Sigil的一些高级功能，这是非常有必要的。

代码视图

要切换到「代码视图」，请单击「代码视图」按钮<>。您可以随时点击「书籍视图」按钮切换回来。或者您可以双击选项卡的标题或使用「视图→切换视图状态」的快捷键。

并列「书籍视图」和「代码视图」进行比较：

introduction.html

```
3    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5  <html xmlns="http://www.w3.org/1999/xhtml">
6  <head>
7    <title>Introduction</title>
8    <link href="../Styles/styles.css" rel="stylesheet" type="text/css"/>
9  </head>
10
11 <body>
12   <h1 id="introduction">介绍 Introduction</h1>
13
14   <p class="h1subheading">— Sigil: EPUB编辑器 —</p>
15
16   <p>欢迎来到Sigil用户指南 - Sigil 0.9.8版。</p>
17
18   <p>Sigil 0.9.8是志愿者经过无数小时努力的结果，创造的最快、最稳定、最强大、版本最好
   的免费EPUB编辑软件。</p>
19
20   <p>跳转到<a href="../Text/tutorials.html">教程</a>，一步步了解如何使用
   Sigil。或查看最新版本<a href="../Text/whats_new.html">更新</a>的功能。</p>
21
22   <p>下载Sigil: </p>
23
24   <ul>
25     <li class="indent"><a href="https://sigil-ebook.com/">Sigil官方网站</
   a></li>
26   </ul>
27
28   <p>在MobileRead中Sigil获得了用户和开发人员的帮助: </p>
29
30   <ul>
```

图5-114　书籍视图
图5-115　代码视图

这是相当多的文字，但它并不像看起来那么乱。首先，忽略文件顶部的所有内容，直到并包括本示例中第11行的<body>标签。「title」只是HTML文件所需的信息，而不是您通常需要更改的内容。

您的文字在<body>标记之后开始：

```
introduction.html
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Introduction</title>
  <link href="../Styles/styles.css" rel="stylesheet" type="text/css"/>
</head>

<body>
  <h1 id="introduction">介绍 Introduction</h1>

  <p class="h1subheading">— Sigil: EPUB编辑器 —</p>

  <p>欢迎来到Sigil用户指南 - Sigil 0.9.8版。</p>

  <p>Sigil 0.9.8是志愿者经过无数小时努力的结果，创造的最快、最稳定、最强大、版本最好的免费EPUB编辑软件。</p>

  <p>跳转到<a href="../Text/tutorials.html">教程</a>，一步步了解如何使用Sigil。或查看最新版本<a href="../Text/whats_new.html">更新</a>的功能。</p>

  <p>下载Sigil: </p>

  <ul>
    <li class="indent"><a href="https://sigil-ebook.com/">Sigil官方网站</a></li>
  </ul>

  <p>在MobileRead中Sigil获得了用户和开发人员的帮助: </p>

  <ul>
```

图5-116　代码视图

您可以看到所有与「书籍视图」相同的单词——章节名称「介绍」和几段文字。但您也可以看到HTML代码标签，这告诉阅读器如何显示该文本。

HTML基础知识

HTML的完整描述超出了本教程，但了解基础知识将有助于您编辑文本。

如果您在「代码视图」中查看文件的HTML代码，您将看到许多单词，如<p>或<h1>等。这些是告诉阅读器如何处理文本的HTML标签。

例如，下面的HTML代码告诉阅读器文本「第8章」是一级标题：

```
<h1>第8章</h1>
```

<h1>标签是被视为标题文本的开头，</h1>标签表示标题中包含的文本的结尾。当您点击按钮 h1 时，Sigil修改您的文档。阅读器通常会使屏幕上的标题文字显示更大。

下面的HTML代码告诉电子阅读器这是一个普通的文本段落。由于它是

一个段落，阅读器将应用一定的格式，也许是首行缩进。

```
<p>这是一段文本。</p>
```

HTML中还有更多的标签。要记住的重点是HTML标签告诉阅读器如何格式化和显示您的内容。它们允许您将内容与样式分开，从而更容易地更改图书的外观。

HTML标签可以有自己的含义，因为它们可以包含被解释的属性。例如，这里是一些段落，其中包含一些标签，用于定义链接的ID，如上一章所示：

```
<p>这是一个<span id="note1">注释</span>链接。</p>
```

`<span>`标签用于需要格式不同或包含信息的小部分文本。在这种情况下，信息是「注释」这个词的ID——「note1」。带有ID属性的`<span>`标签是使用「插入ID」时Sigil插入的代码。

这些只是HTML代码的几个示例，但是它们可以让您了解HTML的大部分内容。

需要记住的是，在「代码视图」中，您可以访问您的文本和控制图书外观的格式代码，以便您可以找到并更改它们。

> 有关详细信息，请参阅简介、代码视图和样式表章节。

在「代码视图」中编辑

一旦进入「代码视图」，就可以像在「书籍视图」一样编辑文本。尝试更改一些单词（但不是HTML标签），然后切换回「书籍视图」。你会看到你所做的修改。事实上，您在「书籍视图」或「代码视图」中所做的任何更改都将在其他视图中显示，因为它们都是完全相同的文件。只是「代码视图」显示了标签，「书籍视图」使用它们来格式化文本。

> 与「书籍视图」不同，您可以在不影响ePub的情况下进行任何更改，当您在「代码视图」中时，必须小心地编辑。您将需要确保您使用的所有标签是有效的，并且正确打开和关闭——如果不是，您将在尝试保存或切换到「书籍视图」时收到警告，您将无法保存您的ePub直到它们被修改正确。

预览

要真正看到代码的修改效果，您可以使用「视图→预览」打开「预览」窗口（您可以随时使用「视图→预览」或其快捷键关闭它）。您可能需要使窗口变大或将其拖动到桌面的另一部分。

「预览」会显示您的文本，如「书籍视图」，但是当您在「代码视图」中时，它将保持打开状态。所以一旦你打开预览，点击「代码视图」按钮<>调出「代码视图」。您可以在「代码视图」中键入文本，修改之后的一秒钟，预览将会更新。



图5-117　代码视图与「预览」窗口

> 经常保存您的代码，及时检查任何错误。

5.4.12 检查拼写错误

Sigil可以使用标准字典和您创建的字典来检查您书中的拼写。

在检查书中的拼写之前，您应该首先使用「首选项→拼写检查字典」配置字典。您将需要选择一个主要词典（英语、法语等）。您可以使用「默认」个人用户字典存储要记住的任何字词，或为每本书创建一个新的用户字典。

拼写检查

要检查您书中的拼写，请单击「拼写检查」按钮或使用菜单项「工具→拼写检查→拼写检查」：

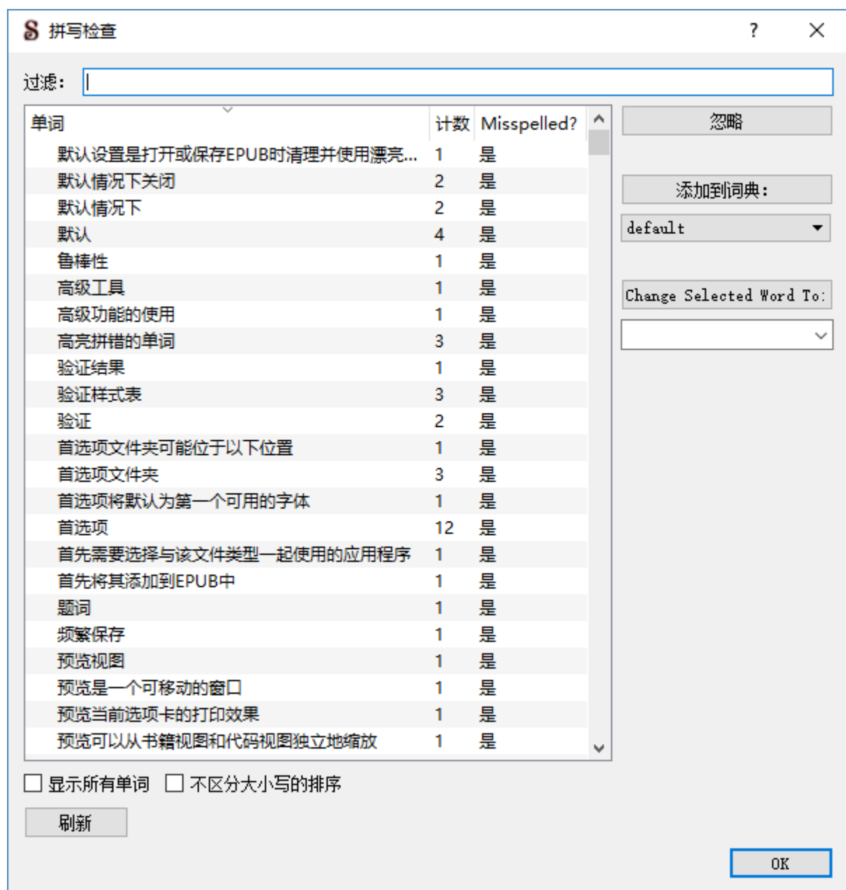


图5-118 「拼写检查」窗口

这会打开一个对话框，显示已经被Sigil识别为拼写错误的单词。

如果您选择一个单词，您将会注意到Sigil会在您的书中找到该单词的第一次出现——并将其显示在代码视图中。这对于在上下文中看到这个词是有用的，但如果您不习惯使用「代码视图」，则可能会有点混乱，但确保Sigil可以准确找到拼写错误的单词。

> 您可以使用「书籍视图」按钮切换回「书籍视图」，但每次选择一个单词时，它将重置为「代码视图」。

一旦你选择了一个单词，你有以下选择：

- 使用忽略可跳过将此单词拼写错误，直到您切换到新书或重新启动Sigil。如果重新启动，Sigil将不会忽略这个词。
- 点击添加到词典按钮，将单词添加到您从按钮下方列表中选择的词典中。从列表中的「默认」字典开始，除非您在「首选项」中创建了其他用户字典。
- 使用Change Selected Word To（更改选择的单词为）将书中所选单词的每一次出现更改为您在下面输入的单词。下拉列表包含建议的替代列表，或者您可以输入您自己的单词。

您可以使用的其他选项有：

- 如果您在拼写检查打开时编辑书籍，则可以使用刷新。如果您只是忽略、添加或更改单词，则不需要使用它。
- 显示所有单词：检查显示在书中所有独特的单词，而不仅仅是拼写错误的单词。
- 不区分大小写的排序：启用此选项可将单词排序为AaBbCc而不是ABCabc。

您还可能会注意到，在「代码视图」中，拼写错误的单词有下划线。这是Sigil检查拼写的另一种方法。

> 有关详细信息，请参阅拼写检查章节。

5.4.13 检查ePub中的错误

你的书几乎已经做好了。您只需检查您的ePub，看看是否有任何可能导致电子书阅读器不能正常工作的问题。这可能包括缺少的或有额外的文件，或链接没有对应上等。对于您的第一个文件可能不太可能，但执行检查是很好的做法，以确保ePub符合标准。

要检查或验证您的书籍，请使用菜单项「工具→Well-Formed Check EPUB」。这将在文档下方的打开一个窗口，并列出发现的任何问题。例如，如果您忘记给HTML标签添加结束标签，您可能会看到：

验证结果

文件	行	Offset	错误信息
OEBPS/Text/Section0001.xhtml	18	473	Error XML not well formed : expected end of tag 'p' near column 3

图5-119 错误信息

备注：0.9.8版本的Sigil验证工具极弱，请使用专业工具来验证你的ePub电子书。另外FlightCrew工具现已做成插件，下载地址。

本教程后面有章节有讲解使用专业的验证工具验证一本电子书的教程，可参考。

有时可能难以解释问题，但您应该阅读它们，看看是否有任何您认识到并可以解决的问题。粉色线是必须修复的错误，黄线是应该修复的警告，有关报错的详细信息，请参见验证一章。

一旦你修复了所有的问题，验证将确认没有发现问题：

验证结果

错误信息

没有发现问题！

图5-120 验证结果

现在只需点击保存按钮，保存您的ePub，就完成了！

继续阅读Sigil的其他功能的教程。

有关如何纠正发现的错误以及如何使用其他工具（如EPUBCHECK）来检查您的ePub，请参阅验证一章。

5.4.14 使用「查找&替换」

在Sigil中，「查找&替换」可用于查找任何文件中的任何文本。更重要的是，它可以用于更改文本，但仅在「代码视图」中。

> 您只能在「书籍视图」中搜索当前文件中的文本。所有其他「查找&替换」操作将切换到「代码视图」中。

要打开「查找&替换」，请使用菜单项「搜索→查找&替换」或其快捷键。「查找&替换」窗口将在编辑窗口的正下方打开，并停留在那里，直到您关闭它：

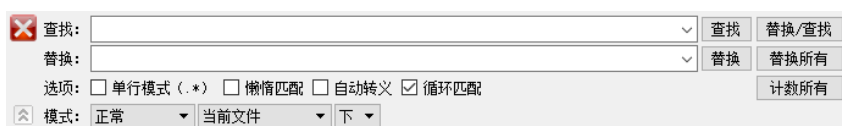


图5-121　「查找&替换」窗口

有很多选项，所以如果你想隐藏其中的一些功能，你可以点击左下方的「显示/隐藏」按钮来隐藏选项行（再次点击则会重新显示）：

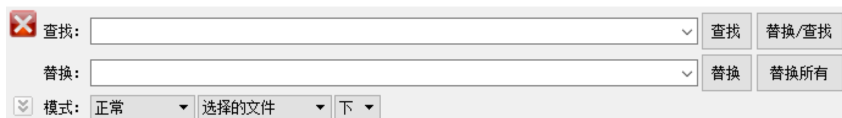


图5-122　「显示/隐藏」选项

> 「查找&替换」文本框是您输入要查找和更改的位置。右侧的按钮（查找、替换、替换/查找、替换所有和计数所有）用于运行搜索。底部的「模式」下拉框允许您更改要搜索的内容、要搜索的位置以及要搜索的方向。
> 每个下拉菜单和按钮都有一个提供更多细节的工具提示。

使用「查找&替换」

要进行简单的更改，请在「查找」框中输入一些要搜索的文本，然后在「替换」框中输入将其替换的新文本：

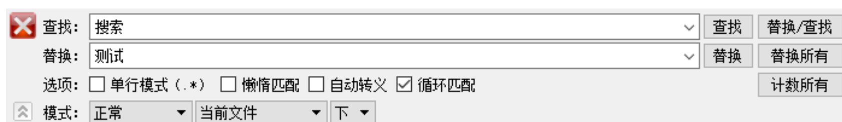


图5-123　替换功能演示

单击计数所有，以显示「查找」框中的文本有多少个匹配项（只计数「查找」，忽略「替换」框中的内容）：



图5-124　计数功能演示

您会注意到，如果您还没有使用它，您切换到「代码视图」！这是因为所有主要的搜索都是在「代码视图」中完成的。您应该留在本教程的「代码视图」中。

请注意，匹配的数量显示在右下方。

现在点击查找。下一个匹配到「查找」框中的单词将突出显示：

```
  <div class="tip">
    <p class="tiptext">You can only search for text in the current file in Book
View. All other Find &amp; Replace actions will switch you to Code View.</p>

    <p class="tiptext">您只能在“书籍视图”中搜索当前文件中的文本。所有其他“查找和替换”操作将
切换到“代码视图”。</p>
  </div>
```

图5-125　查找条目

单击替换——将文本更改为「替换」框中的内容，并使新文本突出显示：

```
  <div class="tip">
    <p class="tiptext">You can only search for text in the current file in Book
View. All other Find &amp; Replace actions will switch you to Code View.</p>

    <p class="tiptext">您只能在“书籍视图”中测试当前文件中的文本。所有其他“查找和替换”操作将
切换到“代码视图”。</p>
  </div>
```

图5-126　替换条目

为了使它更容易的匹配下一个并替换，您可以使用替换/查找按钮。当您点击此按钮时，Sigil将替换所选文本，如果匹配，然后找到并突出显示下一个匹配。

如果您确信替换工作是正确的，您可以使用「替换所有」，一次更新文档中的所有匹配文本，而不是只更改一次。

> 在使用「替换所有」之前，始终使用「计数所有」和可能的几个替换命令，以确保已正确输入「查找&替换」字符串。

您可能已经注意到「当前文件」的选择框。如果将「当前文件」更改为「所有的HTML文件」，则您的「查找」或「替换」将在ePub中的每个HTML文件中完成：



图5-127　查找替换范围

这就是为什么在Sigil中编辑可以快速完成的原因。

「查找&替换」的最后一个主要部分是其使用正则表达式的功能——但是有关更多详细信息，请参阅高级「查找&替换」章节。

5.4.15 使用样式表：使用样式格式化

本教程将介绍样式表以及如何在ePub中使用它们，你也可以参考本书第3章的内容。

样式表是一系列声明，告诉读者如何在书中显示文本。更具体地说，它告诉读者如何使用书中的HTML标签来对文本进行格式化。例如，这是类「author」的段落的样式：

```
p.author {
    font-size: 1.5em;
    font-weight: bold;
    text-align: center;
    margin-top: 4em;
    margin-bottom: 2em;
}
```

图5-128 段落样式

了解和使用CSS，完全按照排版需要用最好的、最灵活的方式格式化你的书籍。

您可以在网上找到其他CSS的介绍，其中包括：

- W3Schools CSS Guide

> 如果您熟悉基础知识，使用CSS就可以参与更多的排版，它将使您更容易了解为什么您的图书看起来是这样的以及如何修改其布局。
> 本章假设您熟悉「代码视图」，因为您需要使用「代码视图」来查看样式的详细信息。

什么是样式？

在书中使用样式做「文本变成蓝色」或「让文本居中」的效果，或者「在这个文本上面留出更多空间」之类的效果。定义样式只需使用相关代码，如下所示：

```
color: blue;
```

这个声明把文字设置成蓝色。

但样式需要在特定标签下才非常有用。将ePub样式表应用于文档中的HTML元素的情况下，如<p>（段落），<h1>（1级标题），<div>（分区），<span>等标签。

在ePub中有3种样式可以使用：

- 样式属性（内联式）：一种直接应用于文档中特定HTML标签的样式。
- 样式元素（<style> </style>）：一个文档中只能由该文档使用的一组样式。
- 样式表：一个单独的CSS或样式表文件，可以被许多HTML文件使用，因此是最灵活的方法。

推荐使用样式表文件方式调用样式。 每种方法的描述将帮助您了解原因。

样式属性

要将样式直接应用于HTML标签，可以使用标签中的style属性。例如，将样式「color:blue;」应用于要更改的段落

```
<p>Hello</p>
```

到：

```
<p style="color: blue">Hello</p>
```

如果您只是想改变一些东西，并且很少更新格式，这个功能很好。但是，如果您以后想要将某些单词的粗体字改为斜体，那么您必须编辑本书中设置特定样式的每个地方。

样式属性很少使用，除非它们是由所见即所得的软件或自动转换创建的，应该避免使用。

有一种更好的方式来管理样式：使用样式选择器。

样式选择器

样式选择器是一种赋予特定样式名称的方式，然后通过引用其名称来使用文档中的样式。更具体地说，样式选择器用于选择文档中的哪些元素应该使用您定义的样式。

例如，如果要将ePub中的所有段落（`<p>`）设置为居中，可以按如下方式为样式选择器定义一个规则：

```
p {
    text-align: center;
}
```

这定义了一个样式选择器p，它将匹配（或选择）所有段落元素p。

只需使用标签名称作为选择器就可以更改每个标签，但也可以选择特定的标签。

选择特定标签的最常见方法之一是在标签和选择器中使用类名。

例如，如果您想要将章节中的第一个段落与其他段落区分，则可以按如下方式定义样式选择器（「em」是一个字符宽度）：

```
p.first {
    text-indent: 0em;
}
```

这定义了一个样式选择器，它首先匹配（或选择）段落标签再使用类名。然后，您可以在文档中使用该样式，方法是将类名称首先分配给段落，如下所示：

```
<p class="first">First paragraph in a chapter</p>
```

这种方式的优点是随时可以更新样式的值（更改缩进或其他声明），而无需更新HTML文本。

> 通常的做法是使用引用样式的类名称，而不是稍后更改样式的内容。例如，如果您希望将地名更改为斜体，请使用span.placename而不是span.bold。

现在你知道一个样式选择器是什么以及如何在你的文本中使用它们，你需要把你的样式选择器放在某个地方才能使它们工作。

在ePub中，您可以将样式表存储在与文本相同的文件（使用样式元素）或单独文件（外部样式表或CSS文件）中。您应该避免将样式存储在HTML文件中，但是如果您在文件中看到这些样式，则将其说明下。

样式元素

如果将样式选择器和声明存储在与文本相同的文件中，则只能将其应用于其所在的HTML文件，而不能被其他HTML文件共享。

使用这些内部样式时，样式定义存储在HTML文件顶部的样式标签中。如果您在书中打开文件并切换到「代码视图」，您将看到文件的顶部包含两个head标签之间的文本。这一部分包含有关文件的信息，您可以在这里放置的其中一个是定义内部样式规则的样式标签。

这是内部样式的典型用法（它们由样式标签包裹）：

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
```

```
<title></title>
<style>
  h1 { text-align: center; }
  p.boldtext { font-weight: bold; }
</style>
</head>
```

内部样式是将样式组合在一起并使用样式选择器的方法，但是当您在ePub中有许多文件时，会变得有点混乱，因为您必须在每个文件中都有单独的样式规则。

管理样式的最佳方法是使用单独的样式表文件来保存样式声明。

> 您可以将内联样式和样式元素转换或复制到单独的CSS文件中。只需记住将您的样式表链接到您的HTML文件，并删除旧的样式属性和元素。

外部样式表——CSS文件

如果创建一个单独的文件来保存所有的样式声明，它们可以应用于整本书。它们也更容易浏览和编辑，更新和管理更容易，因为您添加和删除文件更简单。它也有助于确保您的书籍格式一致，因为您可以在一个地方看到所有的样式。

在Sigil中，外部样式表存储在「图书浏览器」下的「Styles」文件夹中。

要将新的空白样式表添加到您的ePub中，请使用菜单项「文件→添加→空白样式表」：

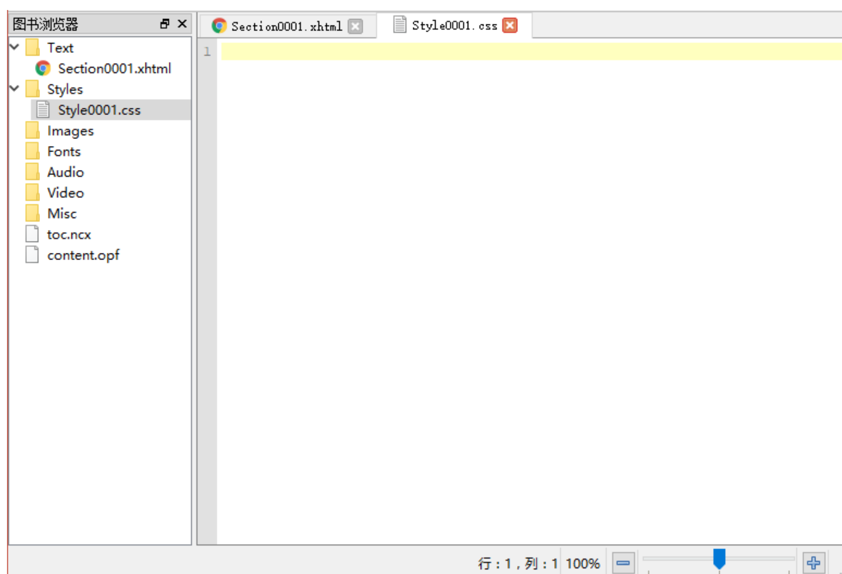


图5-129　空白样式表

你可以看到样式表是空的。

您可以复制或输入所需的样式到文件中，并随时进行编辑。例如，如果您在ePub中打开*Stylesheet.css*文件，您可能会看到如下：

```
h1 {
     font-family: 'SmallCaps', sans-serif;
     font-size: 2em;
     font-weight: bold;
     font-variant: small-caps;
     text-align: center;
     margin-top: 0.0em;
     margin-bottom: 0.5em;
}

h2 {
     font-family: 'SmallCaps', sans-serif;
     font-size: 1.7em;
     font-weight: bold;
     font-variant: small-caps;
     text-align: center;
     margin-top: 0.0em;
     margin-bottom: 0.7em;
}
```

图5-130　样式信息

即使你创建了外部样式表，它实际上并没有被HTML使用。

要使用它，您需要告诉您的HTML文件使用样式表——通过链接到它们，如下所述。

将样式表文件链接到HTML文件中

要告诉您的HTML文件要使用哪个CSS文件，您需要将样式表链接到您的HTML文件：

- 在「图书浏览器」中选择一个或多个HTML文件。
- 右键单击选择操作。
- 从右键菜单中选择「链接样式表」：

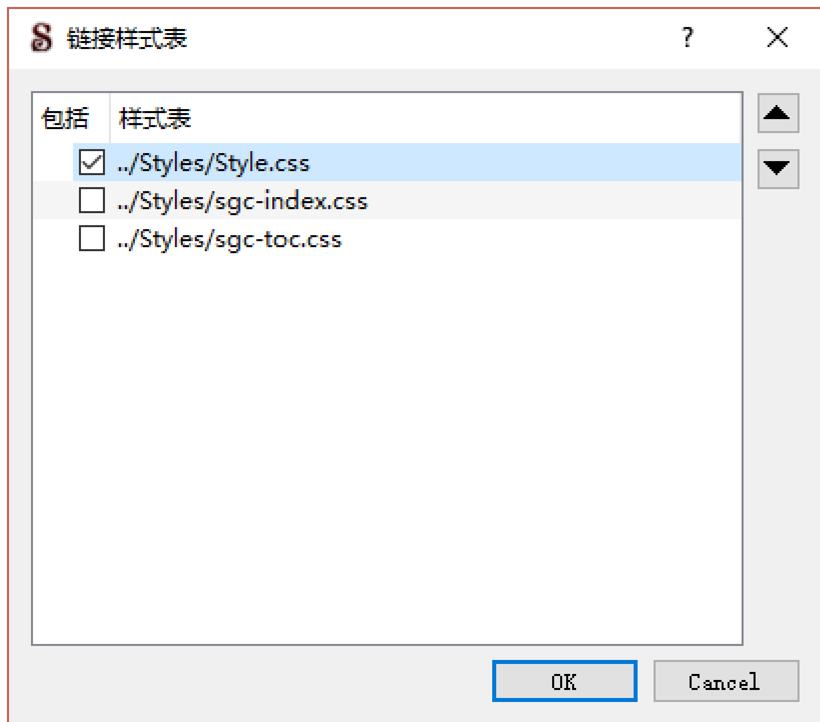


图5-131　链接样式表

- 单击要用于文件的样式表，然后选择OK。

> 如果链接多个样式表，它们将按顺序应用，因此最后一个样式表中的样式将覆盖上述样式表中的样式。

应该使用什么样式？

您用于文本的样式完全取决于您。

学习如何使用样式的最佳方式是找到包含您喜欢的格式的ePub，并查看其CSS文件包含的内容以及它在书中如何使用这些样式。

> 如果您的样式定义有任何错误，则会默认忽略。您可以使用菜单项「工具→验证样式表」来检查错误。

5.4.16 自定义字体：在书中嵌入字体

本教程将向您介绍如何在ePub中使用自定义字体。

> 电子阅读器的一个优点是您可以选择自己的字体，所以通常最好只使用嵌入式字体来突出显示特定类型的文本，例如章节标题，而不是改变整本书的字体。

要在ePub中使用字体，您需要将字体加载到ePub中，为字体定义样式，然后将样式应用于文本。

> 您将需要了解样式表以便您嵌入字体，首先阅读样式表章节。

将字体添加到您的ePub中

要添加字体到您的书籍，请使用「添加现有文件」按钮。这将显示一个对话框，允许您从计算机中选择一个或多个文件。选择尽可能多的字体文件。

> 许多字体不是免费的。在ePub中添加字体之前，请确保您有许可证或有权使用它们。可以在线上搜索查看许多免费字体。
> 字体文件必须是OpenType（.otf）或TrueType（.ttf）格式。所有电子阅读器都应该支持otf，但是大多数可能支持这两种格式。

字体将被添加到「图书浏览器」中的Fonts文件夹。但是，它们还没有在书中的任何地方实际使用——它们只是可以使用。

为您的字体定义样式

一旦您的ePub中的存在字体，您必须定义一个包含它们的样式。

添加或打开链接到HTML文件的样式表（有关详细信息，请参阅样式表一章）。

添加字体的定义，例如：

```
@font-face {
    font-family: "Garamond";
    font-weight: normal;
    font-style: normal;
    src: url("../Fonts/Garamond.ttf");
}
@font-face {
    font-family: "Garamond";
```

```
    font-weight: bold;
    font-style: normal;
    src: url("../Fonts/Garamond Bold.ttf");
}
@font-face {
    font-family: "Garamond";
    font-weight: normal;
    font-style: italic;
    src: url("../Fonts/Garamond Italic.ttf");
}
```

> 字体名称区分大小写！
> 您必须为每种样式的字体（正常、粗体、斜体等）定义一个@font-face规则。
> 当使用font-family的名称时，每次使用字体名称都必须被引用或不引用——您不能使用混合引用和未引用的名称。始终用引号包裹字体名称会更好，检查避免在不使用引号的情况下是否有效。

这本书还没有在书中使用字体，这是下一步。

将您的字体应用于文本

要使用字体，您需要创建另一种样式，然后分配给文本的各个部分。

因此，例如，要使用您的新字体作为标题，请按照以下方式定义标题标签的样式：

```
h1 {
    font-family: "Garamond", serif;
}
```

或某段：

```
p.longhand {
    font-family: "Garamond", serif;
    font-style: italic;
}
```

或者如果你想，虽然不推荐，你可以将字体应用到书中的所有文本：

```
body {
    font-family: "Garamond", serif;
    font-weight: bold;
}
```

> 如果可能的话，上面的font-family规则指示电子书阅读器使用您的字体，但如果不可能则使用通用serif字体。

字体模糊

当您在ePub中使用字体时，该字体文件可能会被别人提取并在别的地方使用。如果您的字体获得许可，您可能希望防止这种情况。

您可以使用Sigil的「字体模糊处理」菜单项，用两种标准方法之一（Adobe或IDPF）来模糊字体文件，使用哪一种方法取决于您，但您应该在目标电子书阅读器上进行测试，以确保字体仍然可读。模糊不是加密——它只是修改字体文件，使其难以在别处使用，同时仍然可以为读者显示字体。

要使用字体模糊，请在「图书浏览器」窗口中右键单击字体，从右键菜单中选择「字体模糊处理」，然后选择一种模糊的方法。

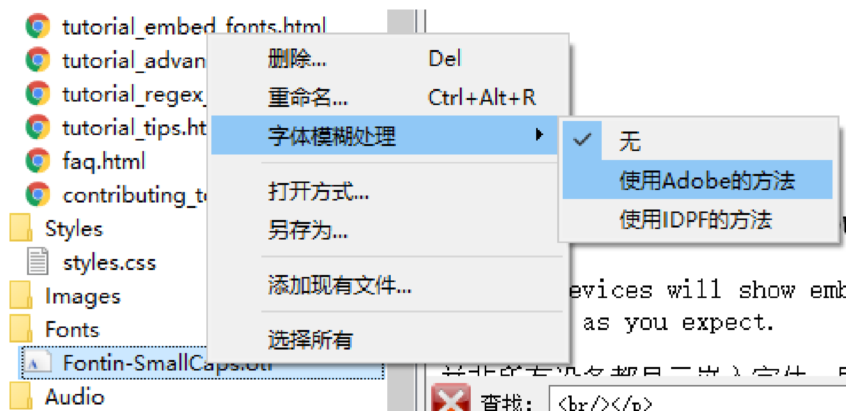


图5-132　字体保护措施

测试你的字体

并非所有设备都显示嵌入字体。即使Sigil也不会显示所有嵌入的字体。最好在阅读器上进行测试，以检查您的字体是否按预期显示。

5.4.17　高级「查找&替换」——正则表达式

通过支持规则表达式（也称为正则表达式），加强了Sigil中「查找&替换」的真正实力。正则表达式基本上是使用模式而不是字面值进行搜索，使匹配文本时更加灵活。

您可以通过将搜索模式从「正常」更改为「正则表达式」，在「查找&替换」中使用正则表达式。

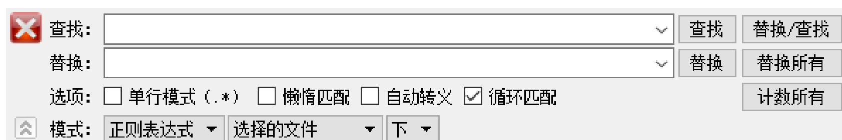


图5-133　「正则表达式」查找

正则表达式是Sigil中非常重要的工具，但在本教程中只能简单介绍一下。有关正则表达式的参考教程，请参阅下一章。（您也可以参考本书中的第4章。）

例如，不是搜索要替换或删除非常具体的文本，例如「Page　123」，搜索包含「Page」后跟数字的文本将更为有用，正则表达式正是可以处理这些情况。

您可以将代替实际文本的正则表达式放入「查找」或「替换」框中。因此，在上面的示例中，您可以搜索「Page　　　　\d+」而不是「Page　123」，因为「\d+」是一个正则表达式代码，意味着任何数字（「\d」）至少重复一次（「+」）。

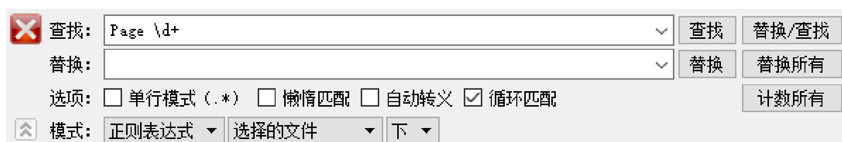


图5-134　正则表达式查找条目

使用正则表达式替换文本

但更有用的是，您可以使用正则表达式来帮助替换文本。在上面的示例中，您可能需要保留页码，而是替换「Page」。您可以使用以下代码：

```
查找：Page (\d+)
替换：This is page number \1
```

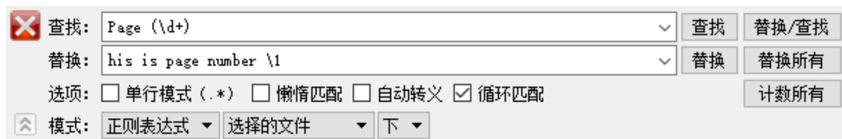


图5-135　使用正则表达式替换文本

这将查找「Page」后的数字，替换到「This is page」后：

```
替换前：Page 123
替换后：This is page 123
```

关于此替换的第一个要点是在「查找」框中使用正则表达式代码`(\d+)`。`\d+`代码告诉「搜索」查找如上所述的数字。但是，在代码周围使用括号会告诉「搜索」记住这些数字——记住括号中匹配的任何数字。另一点需要注意的是在「替换」框中使用正则表达式代码`\1`，该方法告诉「替换」替换查找语句中记住的字符串`\1`，无论它找到什么。

使用正则表达式来更改格式

作为正则表达式的另一个例子，您可以将某些文本的格式更改为章节标题。

假设您有一个已经导入的HTML文件，其中包含许多章节标题，但没有一个使用h1标题标签。相反，它们都被标记为这样的段落：

```
<p> CHAPTER 7 </p>
```

假设这样的每个段落都是章节标题，你可以使用这个正则表达式：

```
查找：<p>\sCHAPTER\s(\d+)\s</p>
替换：<h1>Chapter \1</h1>
```

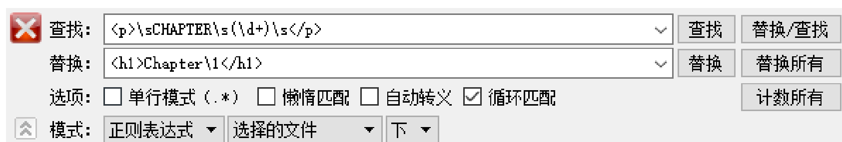


图5-136　使用正则表达式更改格式

这需要理解下，但如果仔细看，你可以看到它非常类似于上面的页码示例，因为它记住了章节名称中的数字，并在替换中使用它们。

这是最有趣的「查找」。分开来解释：

- <p>——寻找起始段落标签。
- \s——以匹配任何空格的正则表达式代码（空格、制表符等）。

- CHAPTER——匹配CHAPTER（正则表达式默认情况下区分大小写）。
- \s——以匹配任何空格的正则表达式代码。
- (\d+)——匹配一行中的任意数字的正则表达式代码，并记住它们。
- \s——以匹配任何空格的正则表达式代码。
- </p>——寻找结束段落标签。

你可以使用一个空格而不是`\s`，但是`\s+`更灵活，因为它将匹配任何数量的空格和制表符。

所以搜索结果可能是：

```
替换前：<p> CHAPTER 14 </p>
替换后：<h1>Chapter 14<h1>
```

> 如果文件中使用的格式不一致，那么使用正则表达式变得更加困难——在这种情况下，您可能需要执行多次「查找&替换」，甚至手动逐个进行修改。

这只是对正则表达式的一个非常基本的描述，但它可以让您了解基础的内容。正则表达式的诀窍是正确找到并替换字符串，这个只有多阅读、提问和测试。

有用的选项

您可能已经注意到「查找&替换」中的额外选项：单行模式、懒惰匹配、自动转义和循环匹配。所有这些选项在使用正则表达式时都很有用。如果您正在开始使用正则表达式，那么建议检查每个框：

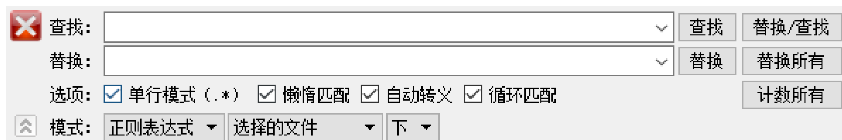


图5-137　「查找&替换」选项

每个选项用于：

- 单行模式：这个正则表达式选项在所有的正则表达式搜索中都会添加`(?s)`，当你想要「.*」时。
- 懒惰匹配：此正则表达式选项对所有正则表达式搜索预先添加`(?U)`，当您希望匹配出现最短的而不是最长匹配时。
- 自动转义：当在所选文本上使用Ctrl+F将文本添加到「查找&替换」时，将会把空格转换为\s并转义某些字符，以便适用于正则表达式。

有关更多详细信息，请参阅正则表达式参考章节。

5.4.18 正则表达式参考

本教程基本上是正则表达式的快速参考。

正则表达式是一种非常强大的工具，您可以在使用「查找&替换」时编辑清理您的ePub。但是，当您第一次使用正则表达式时，可能会令人困惑，因为可以使用很多命令。这是一些基本的正则表达式命令的简要总结。

您可以在这些链接中找到有关正则表达式的更多详细信息（或者在Sigil论坛中请求帮助）：

- Regex overview at MobileRead
- A one page command summary
- Regex as used by Python
- Official PCRE details – start at the PCRESYNTAX(3) section

概述

在「查找&替换」中使用正则表达式命令时，目的是允许您搜索模式相同的文本，而不是精确的文本。这意味着您可以指定「查找&替换」搜索「一个数字」而不是「8」，或者匹配某些标记之间的任何单词，然后用新的标签或新单词替换。

为了允许您搜索模式文本，您需要使用正则表达式命令代替所需的特定文本。例如，不是搜索8，您可以搜索`\d`（意思是一位数字）或`\d+`（表示一个或多个数字）。

一个使用了正则表达式命令「查找&替换」示例，是将所有1级标题更改为2级标题：

```
查找：<h1(.*?)</h1>
替换：<h2\1</h2>
```

其中「查找」命令搜索`<h1>`开始和结束`</h1>`标签之间的任何文本，并使用`()`保存文本。（开始标签为了捕获任何类定义而保持打开状态，并且`.*?`进行最小匹配）。「替换」命令用于将匹配的文本替换为由`<h2>`开始和结束`</h2>`标签包围的保存文本（由`\1`表示）。

以下部分给出了您可能会发现有用的常见正则表达式命令示例。

一般修饰符：

您可以将这些命令放在任何正则表达式的开始处，以修改表达式的行

为：

- (?s)——由于默认情况下`.*`只匹配一行文本。这是通过「查找&替换」中的「单行模式」选项完成的。
- (?U)——匹配您的字符串的第一次出现（也称为最小或不相容匹配）。这在使用`.*`时非常有用，以避免匹配太多。这是通过「查找&替换」中的「懒惰匹配」选项完成的。
- (?i)——匹配时忽略大小写。

您可以组合使用修饰符，例如常用的`(?sU)`。

单字符：

您可以使用这些正则表达式命令来表示单个字符：

- .——匹配一个字符
- \d——匹配一位数
- \t——匹配一个制表符
- [abc]——a b c中的任何一个
- [^abc]——任何不是a b或c的字符
- [0-9]——任何从0到9的数字
- [:alpha:]——任何大写或小写字母
- [^:alnum:]——任何不是数字或大写或小写字母的字符

多个字符：

这些正则表达式限定符或代码允许您表示多个字符：

- *——在一个字符之后使用，表示0或更多，例如.*表示0以上的任何字符
- ＋——在一个字符之后使用，表示1个或更多个，例如\d＋表示1位或更多位数字
- ?——在字符之后使用，表示0或1个实例，例如\d?表示无或1位数
- \s——匹配空格，包括空格、制表符、行尾回车或换行符

还有一个特殊字符，告诉你的搜索停止在它找到的第一个匹配，而不是最大的匹配，你可以使用而不是一般的修饰符`(?U)`：

- ?在量词之后使用最小搜索，例如`.*?`或`\d+?`

分组：

在某些情况下，您需要将您的表达式组合在一起，以便您可以将其保存以供后面替换，或指示单词的选择：

- (atext|btext)——选择atext或btext
- (group)——将这些单词组合在一起以供以后检索——您可以使用多个组

更换：

当替换文本时，最有用的功能之一是能够使用通过将其存储在组中来匹

配的文本。一旦分组文本，您可以在替换命令中引用它：

- \1——在替换中使用以检索保存组的值（对于第二组使用\2等）

5.4.19 提示

以下是检查您的ePub是否完整的一些建议：

- 使用元数据编辑器按钮设置您的标题，作者和语言。如果需要，还可以设置版权、出版日期、主题、介绍和ISBN。
- 为您的封面创建一个HTML文件，并将文件和封面图片设置相应的语义。
- 右键单击您的第一个文本文件，然后选择「添加语义→文本」，告诉电子阅读器这是这本书应该打开的第一页。
- 如果您生成目录，则使用「生成目录」按钮，如果有的话，请重新创建HTML目录，以确保是最新的。
- 测试您的目录链接是否都正确链接了章节。
- 测试您的链接、脚注或尾注是否正确。
- 使用「拼写检查」按钮检查您的拼写，如果它有很多独特的单词，并考虑为书籍创建单独的字典。
- 点击菜单项「工具→Well-Formed Check EPUB」查找ePub错误，然后修复它们——请参阅验证典型错误消息以及如何处理。
- 使用「工具→验证样式表」，检查样式表是否有效。
- 将样式表链接到需要它们的所有文件。
- 使用「工具→删除未使用……」的图像、音频或视频文件或未使用的样式类。
- 运行「工具→报告」并查看任何意外的报告（额外的文件、断开的链接、使用但未定义的类等）。
- 如果您正在使用CSS文件，请搜索内联CSS中使用的「<style>」标签或内联样式使用的「style=」属性，并将其转换为CSS样式表文件。
- 如果您真的很认真，请搜索并替换代码，如 、
、空的段落<p> </p>、空的<div></div>标签，像src="" href=""或=""，更改为–为「to」，除非您希望保留它们。请使用「书籍视图」进行一次检查并再次更新。
- 你甚至可以在代码视图中逐行浏览您的图书，以检查其他错误，特别是如果您已经使用「书籍视图」进行更新。
- 在实际的电子阅读器中测试。Sigil只能给出书的大致浏览效果。尽管ePub是一个标准，但是有很多阅读器设备和软件，他们都有自己的特性和错误。

> 不要忘记——在实际的设备和阅读软件中测试你的书！

5.5 常问问题

本节所列的常见内容均是原英文指南中的内容，译者无增加内容。

5.5.1 常见问题

Sigil是否支持ePub 3？

Sigil是ePub 2编辑器，目前不支持ePub 3的所有功能。但是它支持ePub 3最常用的音频和视频文件功能，并允许您保留ePub 3标签。未来版本可能支持ePub 3，但是ePub 3电子阅读器并不常见，尽管有些电子阅读器支持ePub 3的某些部分。

为什么我的书在Sigil看起来不错，但在我的设备上却不行？

尽管ePub是标准的，但显示您书籍的每个设备或程序使用略微不同的软件，这意味着您的书籍在每个设备上的出现可能会有所不同。Sigil还支持ePub 2标准中的一些功能（如音频、视频和各种HTML 5标签）。在某些情况下，虽然在Sigil看起来还可以，但在特定设备上看起来可能不行。而在其他情况下，它可能在Sigil看起来不理想，但它可能在您的阅读器中显示效果不错。您必须尽可能多地测试您的书籍。

这些「 」单词在我的代码中表示什么？

 是一个空格。如果您在「书籍视图」中进行编辑，则可能会在代码中使用这些HTML字符实体，因为「书籍视图」会在您想要的时候创建它们（例如，如果您在一行中输入两个空格，则将其转换为 才不会丢失——电子阅读器将多个空格减少到一个空格）。如果您的文字实际上并不需要，您可以将其替换为正常字符。

5.5.2 在哪里获得帮助

在哪里可以得到Sigil的帮助？

如果您对Sigil用户指南中没有涵盖的任何关于Sigil的疑问，请访问MobileRead上的Sigil论坛，Sigil的用户和开发人员将尽力帮助您。

在哪里可以询问有关ePub的问题？

您可以在MobileRead的ePub论坛上发布，如果您的问题更多的是如何在ePub中格式化和设计一些东西，而不是在Sigil中做些什么。

哪里可以报告错误或功能请求？

阅读报告问题信息以提交错误报告或功能请求。

5.5.3 使用Sigil

Sigil不能在MAC OS 10.6上启动？

MAC上的Sigil需要MAC OS 10.7或更高版本。

Sigil有教程还是手册？

是的——在Sigil中使用「帮助→用户指南」查看Sigil用户指南的HTML版本，其中还包括教程。或访问Sigil主页以获取HTML和ePub版本的链接。

我可以用Sigil写我的书吗？

是的，你可以，但Sigil主要是编辑一本已经在许多现有文字处理程序中编写的书籍的程序。您可能更喜欢使用您喜欢的编辑器编写，将其导出为HTML，然后使用Sigil将其格式化为ePub。但是，如果你真的想在Sigil写和编辑它，那么你当然可以——《Sigil用户指南》是完全用Sigil写的。

是否支持DOC，Mobi，KF8格式？

Sigil是ePub编辑器。它可以打开ePub文件并导入HTML文件，但是在导入之前，所有其他格式都必须先转换为HTML。

有很多格式，从一种格式转换到另一种格式不是一个小问题。由于已经有一个从一种格式转换为另一种格式的优秀工具，Sigil将继续专注于成为最好的ePub编辑器。

有关详细信息，请参阅准备在Sigil中打开的文件一章。

要将ePub转换为Mobi或KF8，您应该使用Kindle工具。

如何使用正则表达式？

有关使用正则表达式的更多详细信息和提示，请参阅「用户指南」中的「正则表达式」一章。

Sigil不能导入我的文件？

Sigil只能导入或打开有效的HTML和ePub文件。它将尝试修复文件中的一些常见问题，但如果存在重大问题（缺少标准代码或文件），则可能无法打开该文件。

有时Sigil会给你一个错误信息，可能会识别该文件的问题。在这种情况下，您可以使用zip或原编辑程序打开文件，并编辑文件以更正它们。但往往导致问题的原因尚不明确。

如果要报告无法加载和崩溃Sigil的文件的问题，则必须将ePub文件附加到您的错误报告中。您可以将错误报告设置为私有，以便非开发人员无法看到您的文件。但在大多数情况下，您需要更正文件。

拼写检查不正确处理引用或破折号？

如果您安装了自己的hunspell字典，并且它不会识别卷曲引号，em-dashes等，那么您可能需要额外的连字字典文件。有关详细信息，请参阅用户指南中的拼写检查字典首选项。

粘贴到Sigil的文字看起来不对？

将富文本粘贴到Sigil可能无法正常工作。您可以尝试将文本导出（或转换）为HTML，然后直接导入新文件。这可能会保留格式。

如果您要粘贴纯文本，您可能需要代码视图中将其粘贴，以避免保留格式化标记。

5.5.4　格式和样式

为什么我的书显示随机问号？

这可能是因为您的原始文件具有软连字符。ADE（用于在许多系统上显示ePub文件的Adobe软件）不处理。所以只需从文件中删除软连接，问号就会消失。寻找像和‑的。基本上这两个字符之间存在的任何「strange」。您必须使用「代码视图」来查看原始HTML，但您可以使用「报告」功能来显示文档中使用的字符。

也可能是因为ADE无法执行字体替换。

在Sigil中，如果您使用的字体没有所需的字型来呈现文本中存在的字符，则会使用来自不同字体的字型来渲染它。这称为字体替换。所以只

要你的电脑上的某些字体可以显示这些字符，那么Sigil也是如此。

默认情况下，ADE只显示一小部分Unicode字符。支持的字符在Adobe PDF Reference v1.7，附录D，表D.1和D.3中列出。

如果您希望其他字符显示在ADE中，则必须嵌入所需的字体。

如何更改文本的样式？

阅读「用户指南」中的样式表教程——您需要本部分中的许多答案。

请注意，电子阅读器可能支持显示不同的格式或样式，在一个阅读器或Sigil中如何显示文字，可能与其他电子阅读器中显示的方式略有不同。

什么是样式表和CSS？

CSS代表层叠样式表。有关CSS的更多详细信息，请参阅「用户指南」中的样式表教程，以及为什么要使用样式表格式化文本。

可以使用Sigil嵌入自己的字体吗？

可以——请参阅「用户指南」中的字体教程。

为什么我的代码有 和<p> </p>？

如果您在「书籍视图」中进行编辑，则可能会插入空格或其他的格式化文本。「书籍视图」通常用于基本编辑。为了避免意外的字符，您可以在「代码视图」中编辑，或使用「查找&替换」来删除/更改任何意外出现的文本。

如何添加空行？

最好别。更好地的方式是使用样式表定义特殊部分上方或下方的间距。例如，在段落上方留下1.5行空格，您可以使用此样式：

```
p.blankabove {
    margin-top: 1.5em;
}
```

在你的文本中使用你定义的类：

```
<p class="blankabove">My paragraph</p>
```

如果你真的想要，你可以插入一个换行符：

```
<br />
```

或一个空的段落：

```
<p> </p>
```

对于多个空白行，最好使用样式来定义要区分的间距。

如何开始新页面？

如果您为您的页面使用单独的文件，那么几乎所有的阅读器都将为您的文本开始一个新页面。

如何在目录中使用图片作为章节标题？

将您的图像插入您想要的位置——它在「代码视图」中看起来像这样：

```
<img atl="" src="../Images/myimage.png"/>
```

添加要用于图像的标题标签，并将其title显示在目录中，例如：

```
<h1 title="CHAPTER 1"><img alt="" src="../Images/
myimage.png"/></h1>
```

现在目录中将有一个名为「CHAPTER 1」的条目，链接到您的图像。

如果您只想要一个章节条目，您甚至可以使用文档中没有文本的标题标签：

```
<h2 title="Chapter Subsection"></h2>
```

但是，如果图像位于顶部，则可能更容易将目录链接到文件。

如何为我的文本添加页边距？

将您的首选左边距和右边距的代码添加到CSS表中：

```
@page {
    margin-left: 5px;
    margin-right: 5px;
}
```

5.5.5 如何删除段落之间的空格？

要删除所有段落之间的空格，请将其添加到CSS定义中：

```
p {
    margin: 0;
```

```
    padding: 0;
}
```

这将仅影响段落间距（p表示段落）。要删除各种元素之间的所有空格，并使文档看起来更「像书」，请使用以下代码：

```
body, div, p, h1, h2, h3, h4 {
    margin: 0;
    padding: 0;
}
```

如何添加「首字下沉」？

所有阅读器都不支持首字下沉，但有几种方法可以做到首字下沉效果。如果您的书籍格式正确，只需使用CSS即可。

将此代码添加到您的CSS定义中，该定义将使用<span>标签中的dropcap类来设置文本：

```
span.dropcap {
    float: left;
    font-size: 4.7em;
    line-height: .8em;
    margin-right: 3pt;
    margin-bottom: -0.1em;
}
```

在Sigil中，找到要首字下沉的段落，然后切换到「代码视图」。这样改变文字：

```
<p>Hello</p>
```

修改为：

```
<p><span class="dropcap">H</span>ello</p>
```

切换回「书籍视图」，您应该会看到首字下沉效果。

在您想要段落开始处为所有字母添加。

您可能需要调整font-size、line-height和margin-bottom的值以获得所需的效果。Sigil的「书籍视图」可能无法准确显示所有电子书阅读器的显示方式，但相当接近。

为什么我的字体在iBooks中不显示？

如果您在ePub中嵌入字体，则iBooks需要ePub中的非标准文件才能使用您的字体。这在Sigil无法做到。

使用文本编辑器创建名为*com.apple.ibooks.display-options.xml*的文件，并将这些内容放入其中：

```
<?xml version="1.0" encoding="UTF-8"?>
<display_options>
  <platform name="*">
    <option name="specified-fonts">true</option>
  </platform>
</display_options>
```

然后使用压缩软件打开ePub文件。将创建的xml文件添加到ePub中的META-INF文件夹目录下，然后重新压缩ePub。

为什么我的文件由于plist而不通过EPUBCHECK？

如果您在iTunes中打开ePub，它将自动将额外的.plist文件添加到ePub的顶部。这将导致EPUBCHECK失败。如果您打算在其他地方使用它，请勿在iTunes中打开您的ePub。您可以解压缩ePub并删除该文件，然后重新压缩该文件。

5.6 为Sigil做贡献：帮助发展

Sigil是完全免费的开源软件，它完全由志愿者开发和支持，志愿者喜欢书籍，希望编辑好的电子书便于阅读。您可以帮助改善Sigil，并表达您对使软件可用的欣赏。随时在Sigil论坛上发短信，让别人知道Sigil对你多有用。

翻译Sigil

非常鼓励你将Sigil翻译成其他语言。志愿者有一个活跃的社区，他们已经翻译了Sigil，并且可以添加新的语言。

翻译Sigil的最好方法是使用Transifex提供的基于网络的翻译系统。在发布新版本之前下载使用Transifex完成的翻译：

- Sigil translations @ Transifex

或者，您可以使用本地桌面工具进行翻译。Sigil使用Qt的语言学工具来创建和编辑翻译。您可以下载Qt并手动编译工具或使用预构建的软件包：

- Pre-built Translation tools

使用桌面工具更新和创建新的翻译文件是相同的过程：

- 打开终端并切换到src/Sigil/Resource_Files/ts目录。这包含所有原始翻译文件。
- 运行lupdate ../../ * -ts out.ts生成一个新的翻译文件或更新一个现有的添加任何新的字符串。
- 使用Linguist为给定ts文件中的字符串编写翻译。

所有翻译文件应具有sigil_lang.ts格式。其中lang是双字母语言代码。例如，波兰语翻译将具有文件名sigil_pl.ts。

帮助用户

如果您是一名经验丰富的Sigil用户、ePub专家，您可以通过您的经验来帮助回答或评论问题，主要是在MobileRead的官方Sigil论坛上。您还可

以检查开发网站上的问题并添加评论。

编码

如果您是C＋＋程序员，欢迎提交补丁并提出改进建议或错误修复，以供开发者参考。

捐赠

Sigil是免费软件，但非常感谢您的捐款。

请使用「帮助→捐赠」来表示感谢，并帮助维护Sigil。

Sigil用户指南也是免费的。如果你发现它很有用，你可以直接捐赠给作者。

第6章

图像处理——Photoshop

Adobe Photoshop，简称「PS」，是一个由Adobe开发和发行的图像处理软件，是目前公认的最好的通用平面美术设计软件。Photoshop主要处理以像素所构成的数字图像，使用其众多的编辑修改与绘图工具，可以更有效的进行图片编辑工作。

在制作电子书中，主要用Ps来处理封面、插图、生僻字等图片。本章中就讲下Ps的基础用法和一些图片相关知识。至于更高级的操作技巧，可在互联网上自行搜索教程学习。

本章演示所用软件版本为2015版，可能会与你计算机上的界面略有不同。

6.1　图片知识

图片是指由图形、图像等构成的平面媒体。图片的格式很多，但总体上可以分为点阵图和矢量图两大类，本文介绍和比较几种常见图片文件格式的优缺点。

有损与无损

图片文件格式有可能会对图片的文件大小进行不同程度的压缩，图片的压缩分为有损压缩和无损压缩两种。

有损压缩。指在压缩文件大小的过程中，损失了一部分图片的信息，也即降低了图片的质量，并且这种损失是不可逆的，我们不可能从有一个有损压缩过的图片中恢复出全来的图片。常见的有损压缩手段，是按照一定的算法将临近的像素点进行合并。

无损压缩。只在压缩文件大小的过程中，图片的质量没有任何损耗。我们任何时候都可以从无损压缩过的图片中恢复出原来的信息。

索引色与直接色

计算机在表示颜色的时候，有两种形式，一种称作索引颜色（Index Color），一种称作直接颜色（Direct Color）。

索引色。用一个数字来代表（索引）一种颜色，在存储图片的时候，存储一个数字的组合，同时存储数字到图片颜色的映射。这种方式只能存储有限种颜色，通常是256种颜色，对应到计算机系统中，使用一个字节的数字来索引一种颜色。

直接色。使用四个数字来代表一种颜色，这四个数字分别代表这个颜色中红色、绿色、蓝色以及透明度。现在流行的显示设备可以在这四个维度分别支持256种变化，所以直接色可以表示2的32次方种颜色。当然并非所有的直接色都支持这么多种，为压缩空间使用，有可能只有表达红、绿、蓝的三个数字，每个数字也可能不支持256种变化之多。

点阵图与矢量图

点阵图，也叫做位图，像素图。构成点阵图的最小单位是象素，位图就是由象素阵列的排列来实现其显示效果的，每个象素有自己的颜色信息，在对位图图像进行编辑操作的时候，可操作的对象是每个象素，我们可以改变图像的色相、饱和度、明度，从而改变图像的显示效果。点阵图缩放会失真，用最近非常流行的沙画来比喻最恰当不过，当你从远处看的时候，画面细腻多彩，但是当你靠的非常近的时候，你就能看到组成画面的每粒沙子以及每个沙粒的颜色。

矢量图，也叫做向量图。矢量图并不纪录画面上每一点的信息，而是纪录了元素形状及颜色的算法，当你打开一付矢量图的时候，软件对图形象对应的函数进行运算，将运算结果[图形的形状和颜色]显示给你看。无论显示画面是大还是小，画面上的对象对应的算法是不变的，所以，即使对画面进行倍数相当大的缩放，其显示效果仍然相同（不失真）。

6.1.1 格式

ePub支持4种图像格式，具体可参看1.2。本节原文：PNG vs. GIF vs. JPEG vs. SVG - When best to use ?

BMP

BitMap的缩写，是无损的、既支持索引色也支持直接色的、点阵图。

这是一种比较老的图片格式。BMP是无损的，但同时这种图片格式几乎没有对数据进行压缩，所以BMP格式的图片通常具有较大的文件大小。虽然同时支持索引色和直接色是一个优点，但是太大的文件格式格式导致它几乎没有用武之地，现在除了在Windows操作系统中还比较常见之外，我们几乎看不到它。





图6-1　BMP与GIF格式的对比

GIF

全称Graphics Interchange Format，采用LZW压缩算法进行编码。是无损的、采用索引色的、点阵图。

GIF是无损的，采用GIF格式保存图片不会降低图片质量。但得益于数据的压缩，GIF格式的图片，其文件大小要远小于BMP格式的图片。文件小，是GIF格式的优点，同时，GIF格式还具有支持动画以及透明的优点。但GIF格式仅支持8bit的索引色，即在整个图片中，只能存在256种不同的颜色。

GIF格式适用于对色彩要求不高同时需要文件体积较小的场景，比如企业Logo、线框类的图等。因其体积小的特点，现在GIF被广泛的应用在各类网站中。





图6-2　GIF与JPEG格式的对比

JPEG

JPEG是有损的、采用直接色的、点阵图。

JPEG图片格式的设计目标，是在不影响人类可分辨的图片质量的前提下，尽可能的压缩文件大小。这意味着JPEG去掉了一部分图片的原始信息，也即是进行了有损压缩。JPEG的图片的优点，是采用了直接色，得益于更丰富的色彩，JPEG非常适合用来存储照片，用来表达更生动的图像效果，比如颜色渐变。

与GIF相比，JPEG不适合用来存储企业Logo、线框类的图。因为有损压缩会导致图片模糊，而直接色的选用，又会导致图片文件较GIF更大。



图6-3　JPEG与GIF格式的对比

PNG-8

PNG全称Portable　Network　Graphics，PNG-8是PNG的索引色版本。PNG-8是无损的、使用索引色的、点阵图。

PNG是一种比较新的图片格式，PNG-8是非常好的GIF格式替代者，在可能的情况下，应该尽可能的使用PNG-8而不是GIF，因为在相同的图片效果下，PNG-8具有更小的文件体积。除此之外，PNG-8还支持透明度的调节，而GIF并不支持。　现在，除非需要动画的支持，否则我们没有理由使用GIF而不是PNG-8。当然了，PNG-8本身也是支持动画的，只是浏览器支持得不好，不像GIF那样受到广泛的支持。

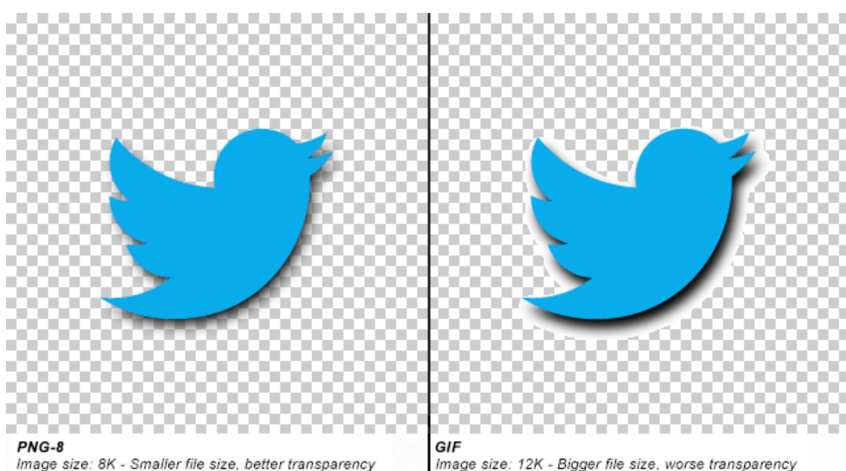


图6-4　PNG-8与GIF格式的对比

PNG-24

PNG-24是PNG的直接色版本。PNG-24是无损的、使用直接色的、点阵图。

无损的、使用直接色的点阵图，听起来非常像BMP，是的，从显示效果上来看，PNG-24跟BMP没有不同。PNG-24的优点在于，它压缩了图片的数据，使得同样效果的图片，PNG-24格式的文件大小要比BMP小得多。当然，PNG-24的图片还是要比JPEG、GIF、PNG-8大得多。

虽然PNG-24的一个很大的目标，是替换JPEG的使用。但一般而言，PNG-24的文件大小是JPEG的五倍之多，而显示效果则通常只能获得一点点提升。所以，只有在你不在乎图片的文件体积，而想要最好的显示效果时，才应该使用PNG-24格式。

另外，PNG-24跟PNG-8一样，是支持图片透明度的。

SVG

全称Scalable Vector Graphics，是无损的、矢量图。

SVG跟上面这些图片格式最大的不同，是SVG是矢量图。这意味着SVG图片由直线和曲线以及绘制它们的方法组成。当你放大一个SVG图片的时候，你看到的还是线和曲线，而不会出现像素点。这意味着SVG图片在放大时，不会失真，所以它非常适合用来绘制企业Logo、Icon等。



图6-5　SVG与PNG-8格式的对比

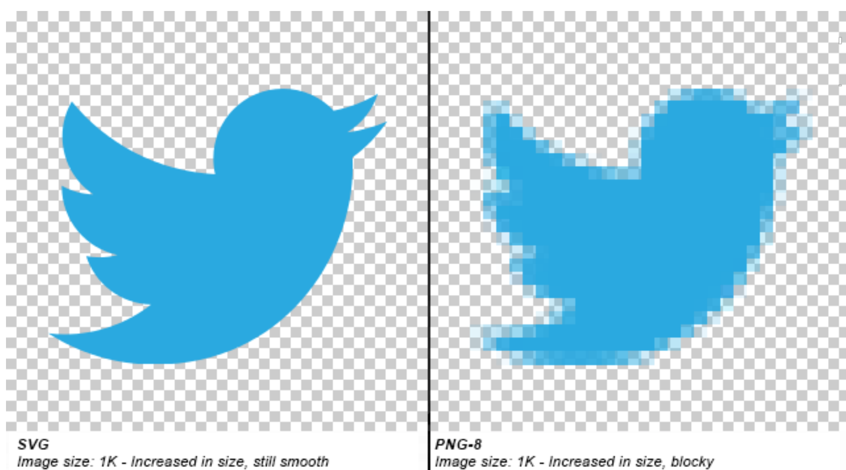


图6-6　SVG与PNG-8的对比（放大）

SVG是很多种矢量图中的一种，它的特点是使用XML来描述图片。使用XML的优点是，任何时候你都可以把它当做一个文本文件来对待，也就是说，你可以非常方便的修改SVG图片，你所需要的只需要一个文本编辑器。

SVG并非只能绘制简单的Logo类的图片，它可以绘制出精致的图片的。

6.1.2　模式

在Ps中保存图片有多种图像格式可供选择，但个别情况下你会发现保存选项中没有JPG了，这是怎么回事？今天就此问题，讲解一下图像模式的区别与JPG的关系。

RGB模式

首先打开一张图片，然后依次点击「图像→模式」，图像显示为「RGB颜色」选项。这是Ps常用的色彩模式，也称为「真彩色模式」。

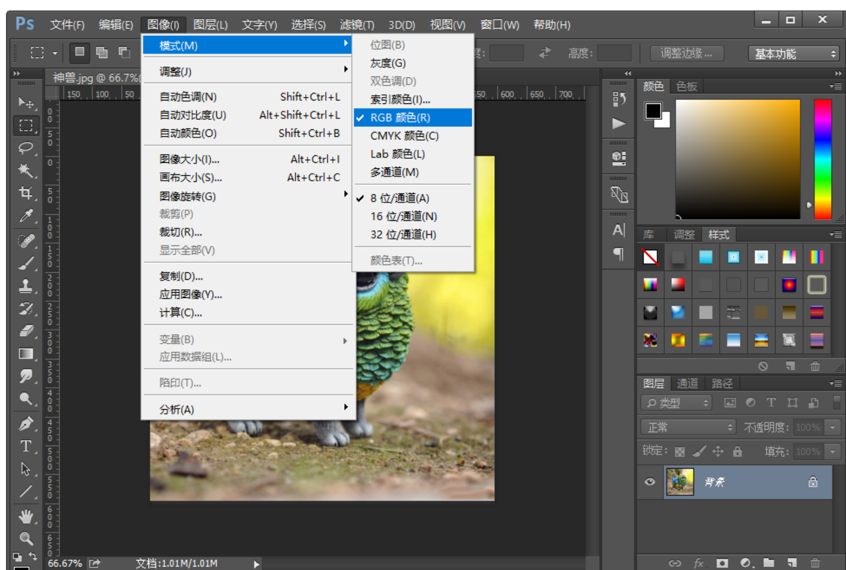


图6-7　设置图像为RGB颜色模式

另外，点击通道按钮，可发现通道中除RGB通道外，有红、绿、蓝三个通道。

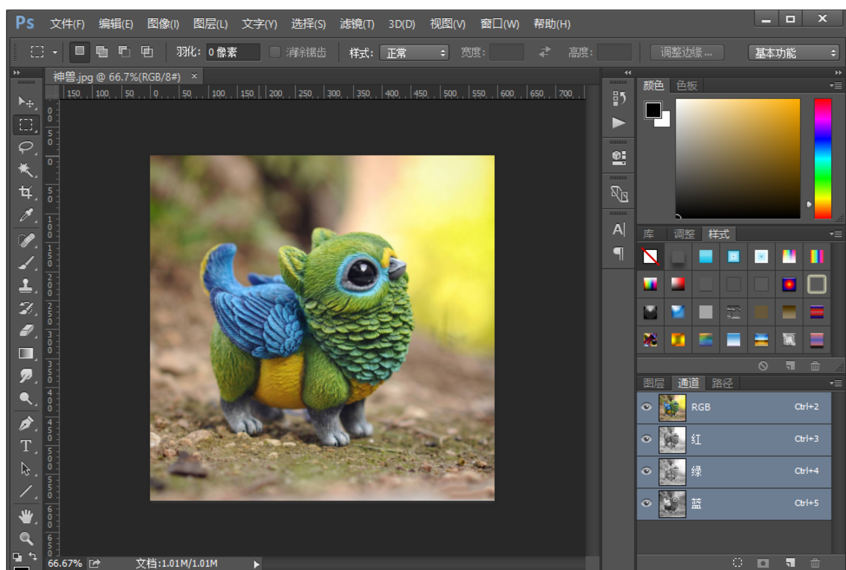


图6-8　RGB图像通道

即使编辑完图像，点击「存储」，也会有很多格式类型进行选择，如图：

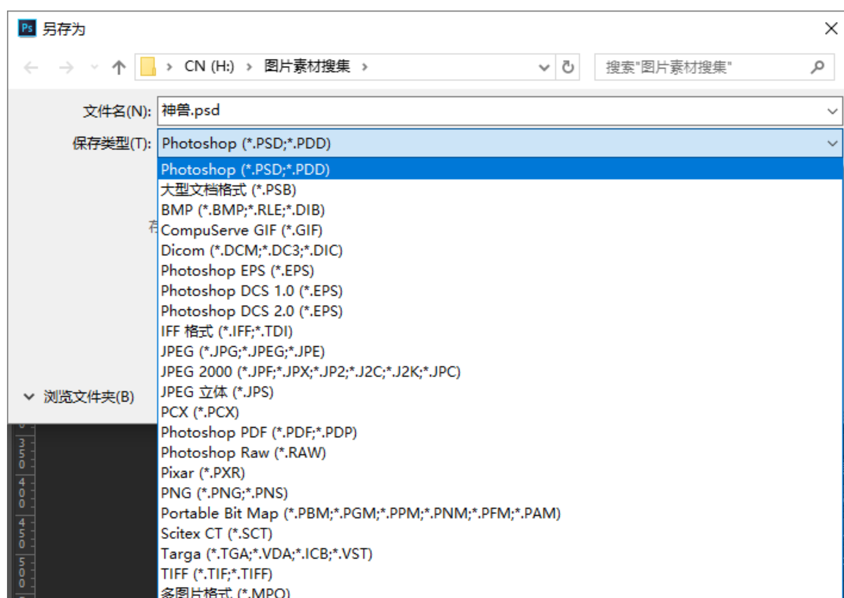


图6-9　RGB颜色模式的保存格式

> 以RGB模式保存的图像质量最高，一般为Photoshop默认模式。

在3.4.2 背景属性一节中对RGB有描述，可查阅。

CMYK模式

此模式也是PS的常用模式，一般为印刷使用。

印刷四色模式是彩色印刷时采用的一种套色模式，利用色料的三原色混色原理，加上黑色油墨，共计四种颜色混合叠加，形成所谓「全彩印刷」。四种标准颜色是：C：Cyan＝青色，又称为「天蓝色」或是「湛蓝」；M：Magenta＝品红色，又称为「洋红色」；Y：Yellow＝黄色；K：Key Plate＝定位套版色（黑色）。

它和RGB相比有一个很大的不同：RGB模式是一种发光的色彩模式，你在一间黑暗的房间内仍然可以看见屏幕上的内容；CMYK是一种依靠反光的色彩模式，我们是怎样阅读纸书的内容呢？是由阳光或灯光照射到纸书上，再反射到我们的眼中，才看到内容。它需要有外界光源，如果你在黑暗房间内是无法阅读纸书的。

例如将RGB图片转为CMYK时，依次点击「图像→模式→CMYK　颜色」：

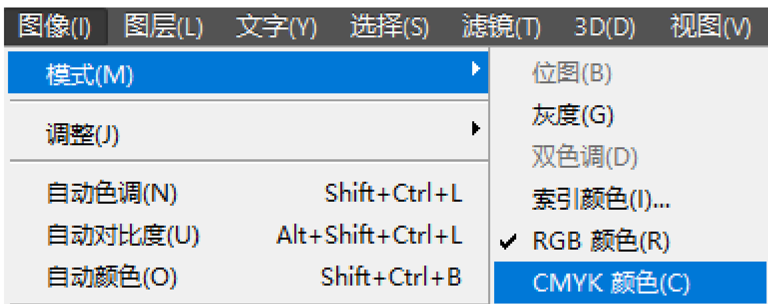


图6-10　设置图像为CMYK颜色模式

提示合并图层信息，并进行转置。



图6-12　配置CMYK

此时，通道中，按CMYK模式C/青色、M/洋红、Y/黄色、K/黑色进行显示：

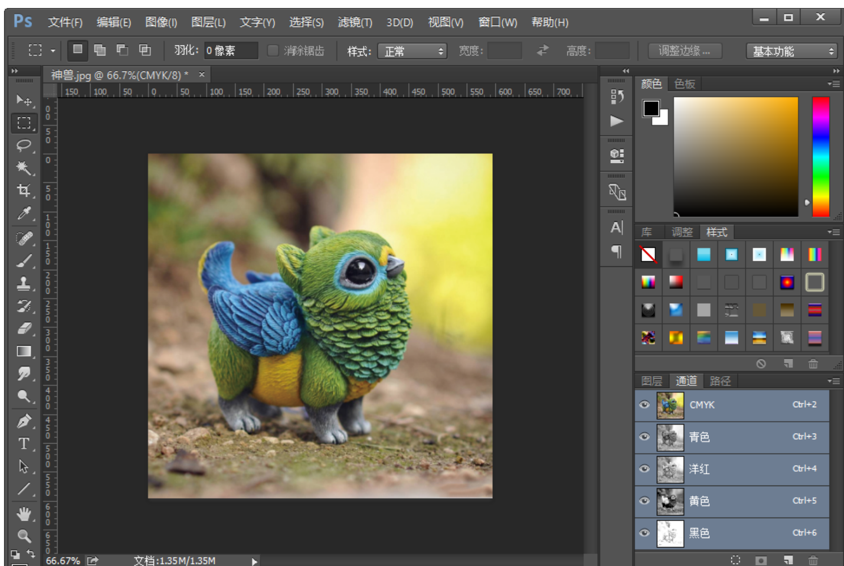


图6-13　CMYK图像通道

点击「存储」时，选项类型如下，主要是JPG格式类型。

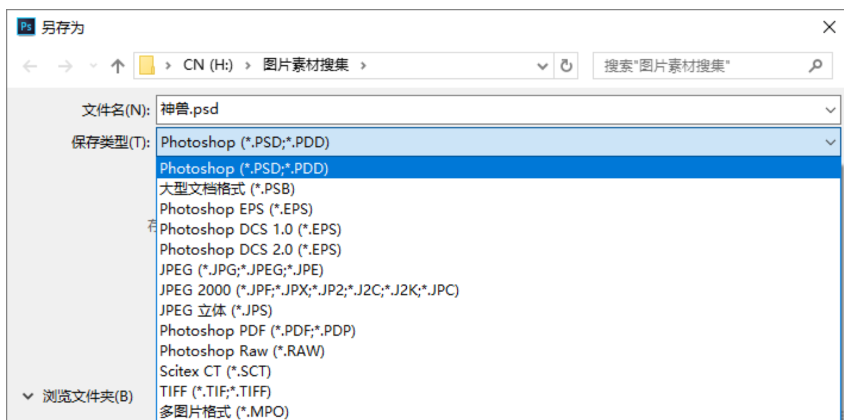


图6-14　CMYK颜色模式的保存格式

索引颜色

索引颜色是位图图片的一种编码方法，需要基于RGB、CMYK等更基本的颜色编码方法。可以通过限制图片中的颜色总数的方法实现有损压缩。

挑选一副图片中最有代表性的若干种颜色（通常不超过256种），编制成颜色表。在表示图片中每一个点的颜色信息时，不直接使用这个点的颜色信息，而使用颜色表的索引。这样，要表示一幅32位真彩色的图片，使用索引颜色的图片只需要用不超过8位的颜色索引就可以表达同样的信息。使用索引颜色的位图广泛用于网络图形、游戏制作等场合，常见的格式有GIF、PNG等。

当RGB图片转为索引颜色时，也会提示拼合图层：

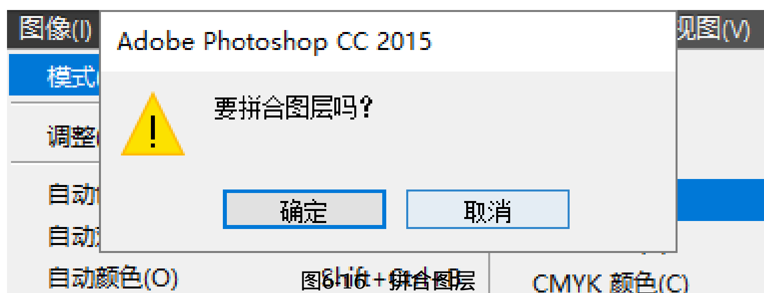


图6-15　设置图像为索引颜色模式

在第二个对话框中，会要求修改相应的数值。

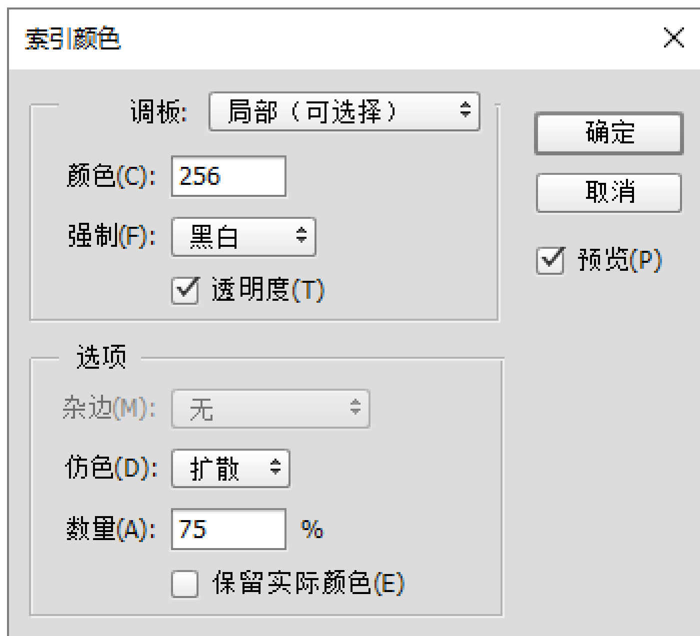


图6-17　索引颜色设置窗口

确定后，可以看到图片颜色损失严重。在通道中，也只显示一个索引项目。

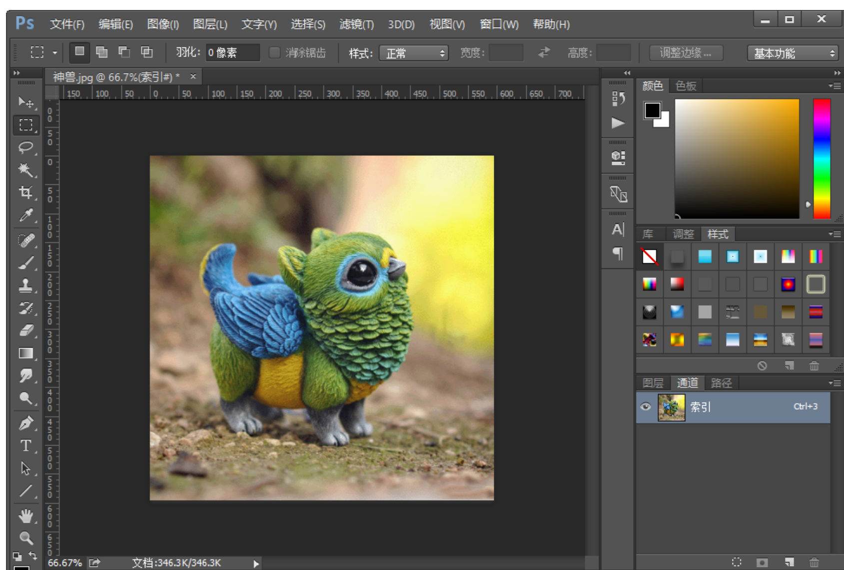


图6-18　索引颜色通道

而且，存储时，选项明显减少；可以看到没有JPG格式。相应的GIF，PNG有所保留。

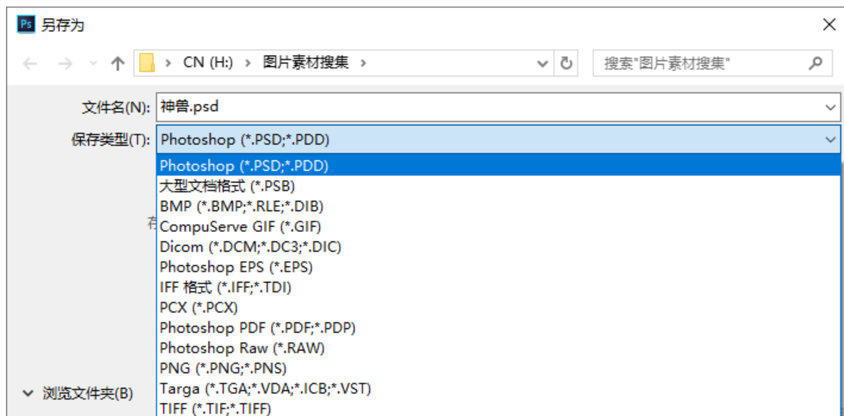


图6-19　索引颜色模式的保存格式

Lab颜色模式

Lab颜色模型基于人对颜色的感觉。Lab中的数值描述正常视力的人能够看到的所有颜色。因为Lab描述的是颜色的显示方式，而不是设备（如显示器、桌面打印机或数码相机）生成颜色所需的特定色料的数量，所以Lab被视为与设备无关的颜色模型。

Lab色彩模型还具有色域宽阔的优势，它不仅包含了RGB，CMYK的所有色域，还能表现它们不能表现的色彩。

依次点击「图像→模式→Lab 颜色」，会提示拼合图层：

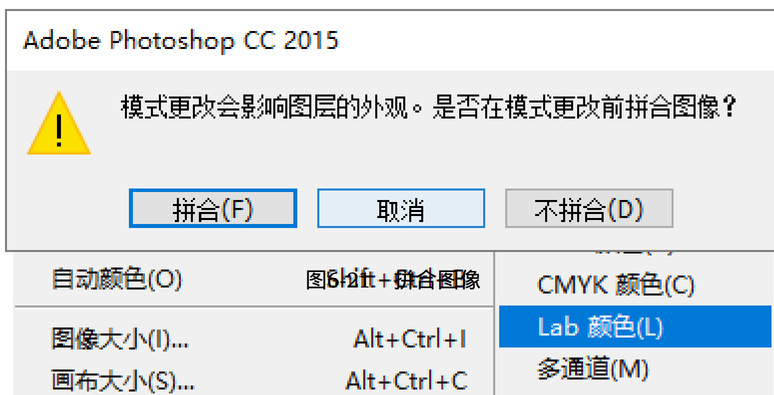


图6-20　设置图像为Lab颜色模式

点击「拼合」按钮后会直接应用，且颜色效果无异：

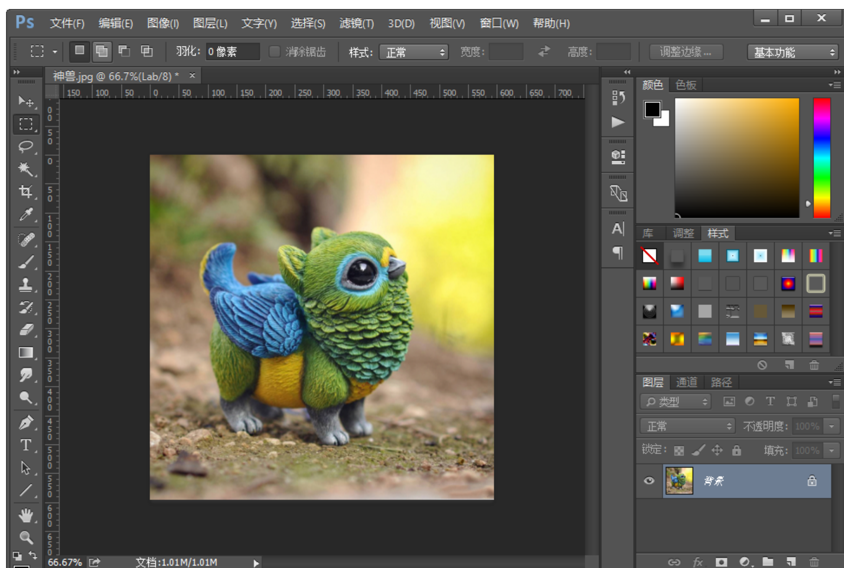


图6-22　Lab颜色模式的预览

在通道中，以明度（L）、a（由绿到红）、b（由蓝到黄）三个通道显示/构成。

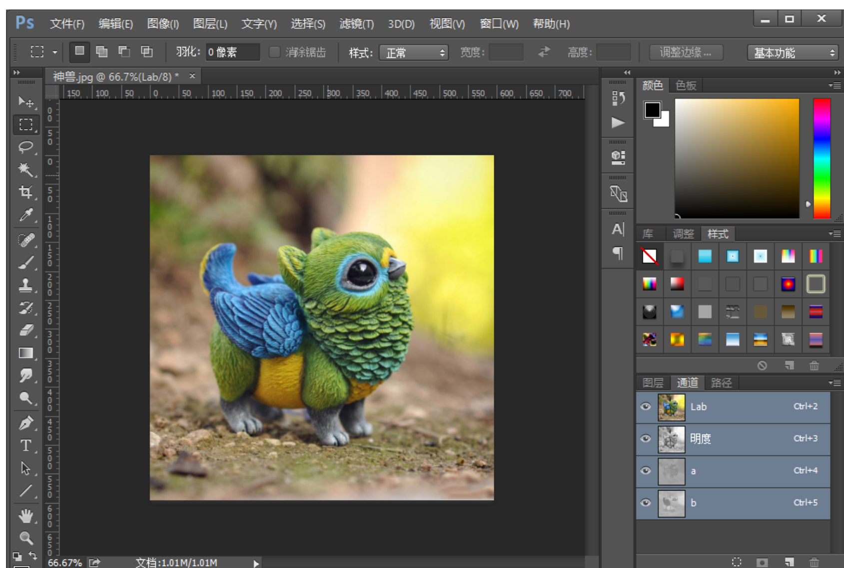


图6-23　Lab颜色模式的通道

其次，在保存Lab颜色模式时，存储选项会更少。基本只以PS及JPG格式保存。

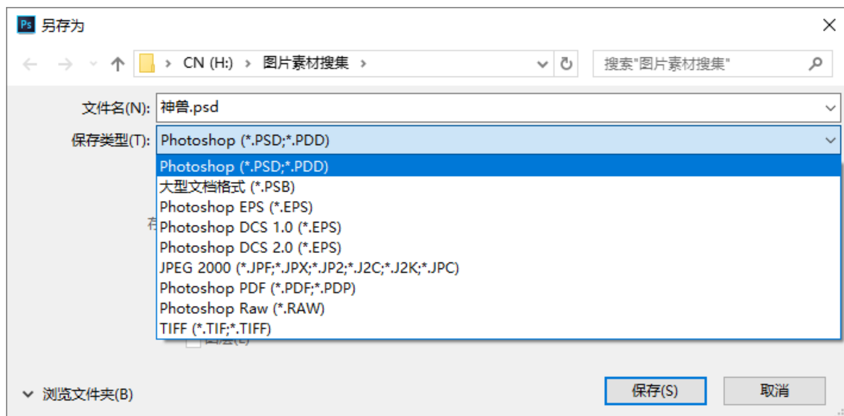


图6-24　Lab颜色模式的保存格式

灰度模式

所谓灰度色，就是指纯白、纯黑以及两者中的一系列从黑到白的过渡色。灰度色中不包含任何色相，即不存在红色、黄色这样的颜色，每个像素以8位或16位颜色表示。

依次点击：

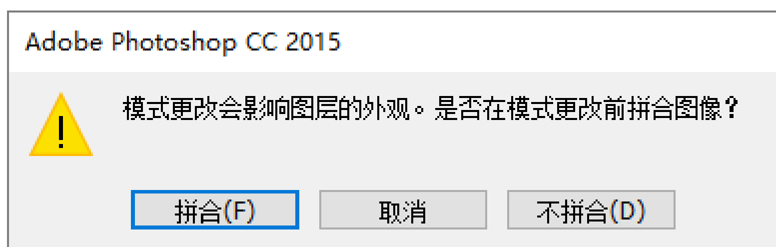


图6-26　拼合图像

点击「拼合」按钮后，提示扔掉颜色信息：

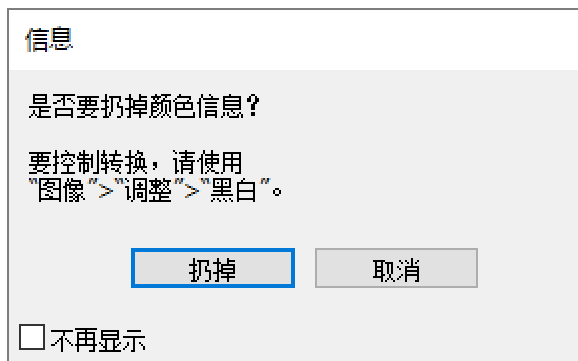


图6-27　扔掉颜色信息提示

将彩色图像转为灰度模式后，所有颜色会被不同的灰度代替。且在通道中，只有一种颜色——灰色。另外，在「存储」时，主要类型与原RGB存储选项相同。



图6-28　灰色通道

多通道模式

英文又称：Multichannel。在多通道模式中，每个通道都含用256灰度级存放着图像中颜色元素的信息。该模式多用于特定的打印或输出。一般包括8 Bits channel（8位通道）与16 Bits channel（16位通道）。

通过从RGB、CMYK或Lab图像中删除一个通道，可以自动将图像转换为多通道模式。

在使用多通道模式下，「存储」时只有4个选项。如图：

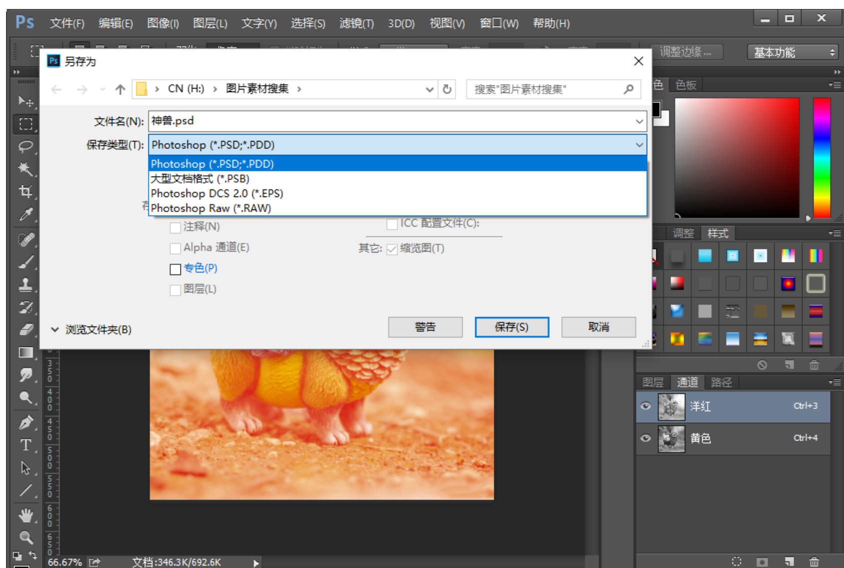


图6-29　多通道模式的保存格式

6.2　使用「动作」处理图片白边

如果从某一种电子书格式转换后的ePub文件的插图有不少白边的话，可以手动在Ps中修改。但是遇到插图特别多，几十几百张，一张张修改的话非常的费时费力。Ps中的「动作」功能可以帮助我们解决类似的情况，它的原理和搜索模板差不多，都是预先设定每一步做什么，由软件自动处理每一步，制作者只需要设定好输入输出参数就行。

录制动作

在修改ePub电子书的过程中，有时会遇到给图片去除白边的需求。比如下面的这种：

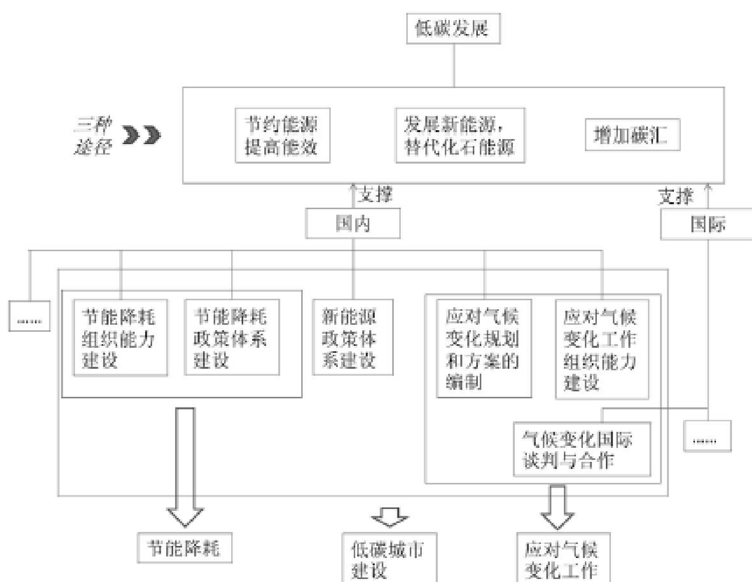


图6-30　待裁切的图片

你可以保存上面的图片，然后跟着下面的步骤操作，录制自己的去白边动作。

本次教程使用的Photoshop版本是2015版，界面可能与你使用的不同。

依次点击「文件→打开」或使用快捷键Ctrl+O，在弹出窗口中选择要去白边的图片，然后点击「打开」按钮。依次点击「窗口→动作」或使用快捷键Alt+F9调出「动作」面板，点击红色箭头所指的图标，新建动作。在弹出对话框中输入动作名称，点击「记录」按钮：

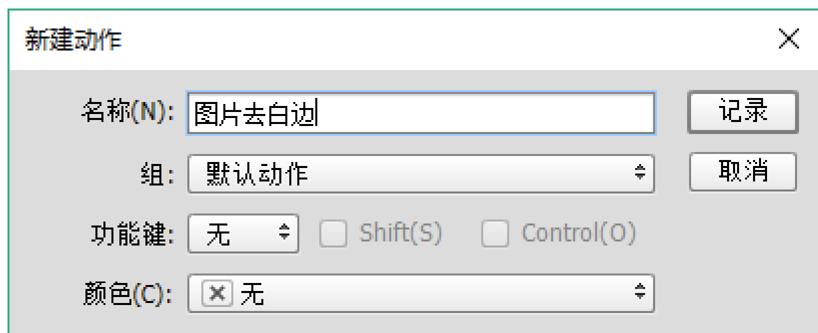


图6-32　重命名动作

图6-31　新建动作

「动作」面板会开始记录之后的所有操作，此时面板如下：

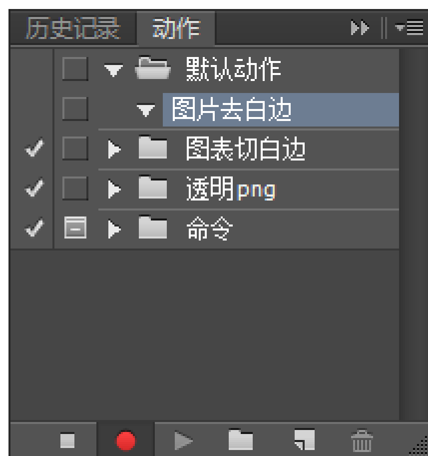


图6-33　录制操作记录

依次点击「选择→色彩范围」，在弹出窗口中的「选择」下拉菜单中选择「高光」，颜色容差调为0,、范围252，勾选「反相」：



图6-34　设置色彩范围

点击「确定」按钮，保存退出当前窗口。在Ps主界面中点击左侧工具栏「裁剪」按钮，或者把输入法调为英文/关闭状态，按C键。然后按两次Enter键，确认裁剪操作。依次点击「文件→存储」或使用快捷键Ctrl＋S保存修改，依次点击「文件→关闭」或使用快捷键Ctrl＋W关闭当前标签。点击红色箭头所指的图标，结束动作的录制。

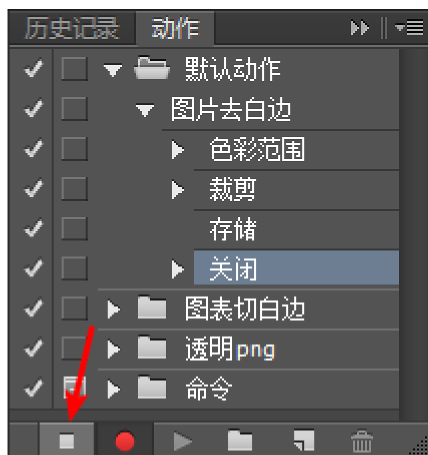


图6-35　结束动作的录制

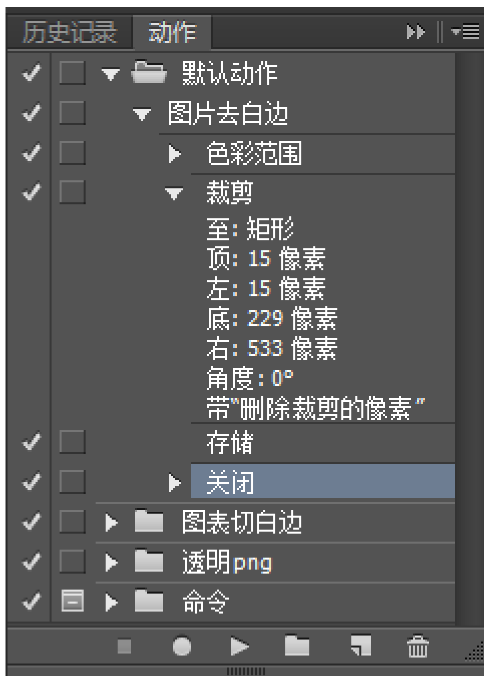


图6-36 裁剪的坐标信息

录制好的动作现在还不能用，因为这个动作只能处理我们刚才演示的那张图片，它是固定了坐标的，见上面的右图。选择「图片去白边」里的「裁剪」动作，点击右上角的菜单图标，点击「命令」选项。下拉找到「裁剪（选区）」选项，点击左侧的三角形展开它，按住Alt键+鼠标左键向上拖动「裁剪」到「图片去白边」动作里。

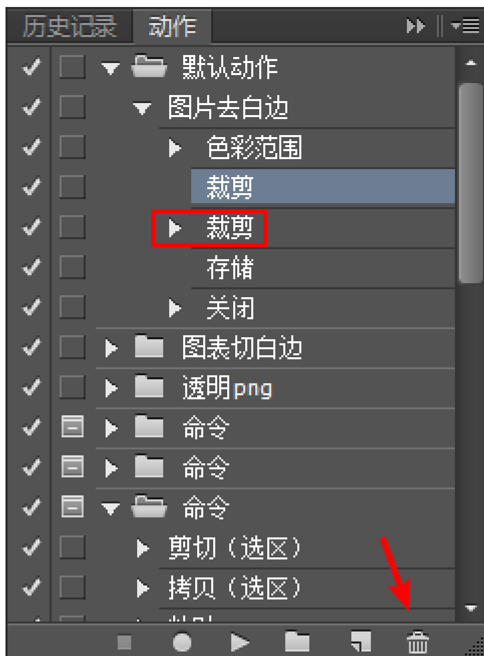


图6-37　修改裁剪参数

选择上图红框中的「裁剪」，点击红色箭头所指的「删除」按钮，删除它。完成后，可以打开上面3张中的图片任意一张进行测试，步骤参考下面的内容。

录制好的动作，可以保存到计算机上进行分享，点击动作面板右上角的「菜单」图标，选择「存储动作」选项。在弹出窗口中，输入动作名称，点击「保存」按钮。

单张图片处理

下面就是一张上下带有白色背景的图片：



图6-38　待裁切的图片

先下载一个图表切白边.atn动作文件。然后打开你的Ps软件，按「Alt＋F9」键，弹出「动作」面板，如下：

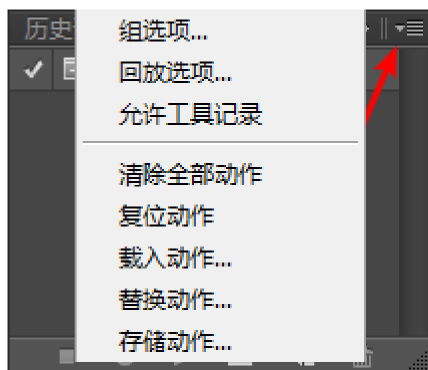


图6-40　选择「载入动作」选项

图6-39　载入动作

点击红色箭头指向的位置，弹出窗口（部分）如上面右图。

选择「载入动作」按钮，在「打开」对话框中，找到动作存放的文件夹，选择文件，点击「打开」按钮，此时Ps中的「动作」面板如下：

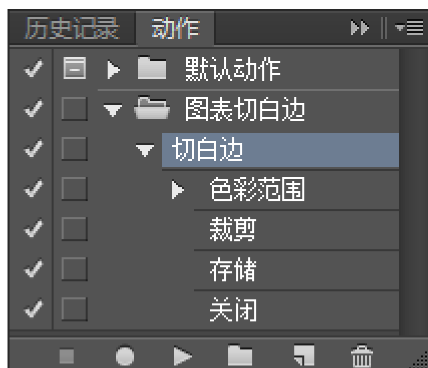


图6-42　选择需要进行的操作

选择「切白边」组，底部三角形变成可点击状态。然后打开你需要处理的图片，点击黑色三角形按钮，播放动作。完成后即可看到图片已经被处理了。效果如下：



图6-43　裁切好的图片

批处理多张图片

多张图片的处理需要用到Ps中的「批处理」功能。依次点击「文件→自动→批处理(B)...」（Photoshop 2015版本）：

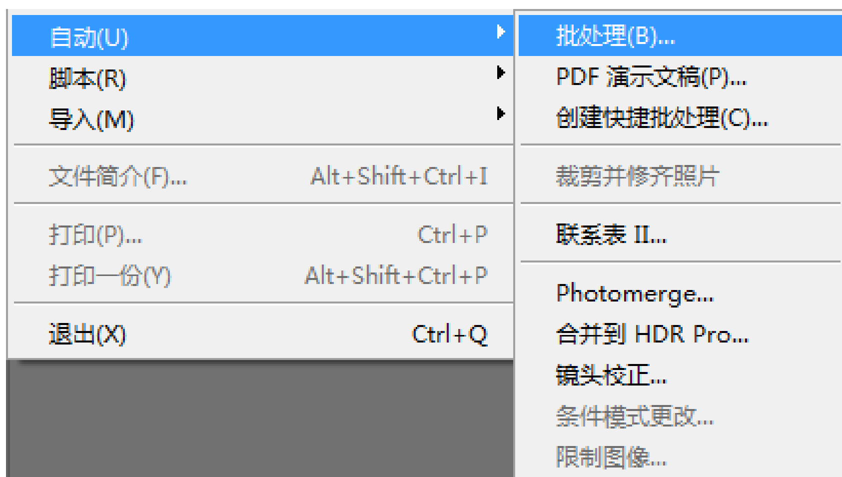


图6-44　使用「批处理」功能

然后会弹出下面的对话框，这里可以设置使用的动作、输入输出文件夹等参数。



图6-45　设置「批处理」参数

「播放」设置项中，组选择「图表切白边」、动作选择「切白边」，源设置为「文件夹」、点击「选择」按钮选择需要切白边的图片所在文件夹。在处理之前，建议先备份一个文件夹，切白边成功后，再删除备份。目标设置成「存储并关闭」，然后点击右上角「确定」按钮。

Ps会打开一张张图片，并依次执行动作。直到全部图片处理完毕，处理完毕的图片，建议检查下。

6.3　一张可用的封面

我们在做电子书中，有时会遇到没有封面、只有立体封面或只有小图的情况。遇到这种情况时，第一种方法是在全国图书馆参考咨询联盟中下载辅助页，从中提取出封面图片。这种方法的缺点是，封面图片因为是扫描的，并不是非常清晰，有点模糊（经过压缩）。第二种方法是使用Ps软件，对立体封面进行修改，以达到可使用的目的。第三种方法是根据找到的封面小图，在Ps中进行重制（前提是设计并不是很复杂）。

电子书的显示需要借助阅读软件来显示内容，而不同的阅读软件运行的硬件又不一样。总体来说分为平板和手机两种屏幕尺寸。

- 以iPad为例，常见尺寸有1024×768、2048×1536两种，比例是4:3；
- 以Android手机为例，常见尺寸有1280×720、1920×1080两种，比例是16:9；
- 以iPhone手机为例，常见尺寸有1334×750、1920×1080等，比例是16:9；

这样，我们在制作电子书时，只需要把封面处理成两种尺寸比例的即可，一种是4:3，一种是16:9。为了在高像素密度（PPI）的屏幕中也有较好的显示效果，我们把封面的尺寸设大一些，确定为高2048×宽1536（平板）、高1920×宽1080（手机）两种。这样在读者使用平板设备阅读本书时，调用平板尺寸的封面；使用手机阅读时，调用手机封面。这里涉及到每个阅读器的私有属性，在后面排版样例中会讲。

最近手机圈又出现一种「全面屏」技术的手机，可视区域的比例达到了18:9，所以以后做电子书封面还需要考虑这种手机。

6.3.1 图书馆中获取封面

在网络中搜索一本书的封面，一般现在搜索引擎中根据书名或出版社等关键字搜索，从搜索结果中找到可以使用的封面图片。图书电商网站推荐亚马逊，如果搜索结果和亚马逊都没有满意的图片，则在全国图书馆参考咨询联盟中搜索。

汇集信息

确定书名、作者、出版日期、出版社，这一步是确定所做电子书的版本。下面以《中国小说史略》为例，这本书有很多家出版社出版过，同一出版社又有不同的版次。

书名：中国小说史略
作者：鲁迅
出版社：上海古籍出版社
出版日期：2006.4

搜索书名

根据以上信息，我们在全国图书馆参考咨询联盟中搜书名。首先打开这个网站，在输入框中输入书名：

全国图书馆参考咨询联盟

图书 期刊 报纸 学位论文 会议论文 专利 标准 音视频
中国小说史略 中文搜索 外文搜索 高级检索 使用帮助
全部字段 书名 作者 主题词 丛书名 目次

图6-46 输入搜索信息

搜索类型限定为「图书」，然后点击右侧的「中文搜索」按钮。这回跳转到新页面，展示一些搜索结果：

类型
在线试读(68)
年代
2017(1)
2016(3)
2015(5)
2014(10)
2013(4)
2012(7)
2011(8)
2010(7)
2009(4)
2008(3)
更多
学科
文学(92)
综合性图书(2)
自然科学总论(1)
作者
鲁迅(82)
郭希汾(3)
陈淑渝(1)
郭箴一(1)
储大泓(1)
方正耀(1)
更多..

《中国小说史略》
作者:鲁迅著 页数:365 出版社:北京:中国书籍出版社 出版日期:2016.06
丛书名称:中国史略丛刊
简介:本书共分为二十八篇，主要内容包括:史家对于小说之著录及论述、神话与传说、
主题词:鲁迅著作-小说史-中国

《中国小说史略》
作者:鲁迅 页数:336 出版社:南宁:广西人民出版社 出版日期:2017.01

《中国小说史略》
作者:鲁迅著 页数:253 出版社:南京:译林出版社 出版日期:2015.01
丛书名称:字里行间文库
简介:本书主要是研究中国古代小说发生、发展、演变的过程，将各类型小说及其发展
主题词:鲁迅著作-小说史-中国

《中国小说史略》
作者:鲁迅著 页数:236 出版社:北京:民主与建设出版社 出版日期:2015.05
简介:《中国小说史略》是一部古典小说史名著，是鲁迅先生1920年至1924年在北京大
国小说史的开山之作，首篇为《史家对于小说之著录及论述》，艺术技巧，演变趋势等
主题词:鲁迅著作-小说史-中国

图6-47　搜索结果列表

核对版本

点击左侧的「更多」按钮，再选择「2006」年份，然后会出现筛选结果，根据书的版本信息选择正确的搜索结果：

类型
在线试读(3)
年代
全部>2006
学科
文学(4)
作者
鲁迅(3)
陈淑渝(1)
聂石樵(1)

《中国小说史略》
作者:鲁迅著 页数:196 出版社:上海:上海古籍出版社 出版日期:2006.04
丛书名称:世纪文库
简介:在中国历来鄙视小说的氛围中，鲁迅先生的《中国小说史略》率先把小说作
迅先生为撰写《中国小说史略》作了大量的史料勾稽、考证工……
主题词:鲁迅著作 学科: 小说史 学科: 中国

《中国小说史略》
作者:鲁迅著 页数:348 出版社:北京:人民文学出版社 出版日期:2006.12
简介:0903050101
主题词:鲁迅著作-小说史-中国

图6-48　选择需要的搜索结果

第一个筛选结果就是我们要找的书目信息。点击封面或书名，进入书籍详情页。在右侧的「获取途径」中发现有试读选项，点击「版权页」或「前言页」等链接：

中国小说史略　　　　试 读



【作　者】鲁迅著
【丛书名】世纪文库
【形态项】196
【出版项】上海：上海古籍出版社，2006.04
【ISBN号】7-5325-4320-X
【中图法分类号】I210.91
【原书定价】25.00
【主题词】鲁迅著作(学科：小说史 学科：中国)
【参考文献格式】鲁迅著．中国小说史略．上海：上海古籍出版社，2006.04.

内容提要：

在中国历来鄙视小说的氛围中，鲁迅先生的《中国小说史略》率先把小说作为一门专门的学问加以系统的研究，开创了中国古代文学研究的新领域，建立了中国小说史的独立体系，为中国小说史的研究奠定了坚实的基础。鲁迅先生为撰写《中国小说史略》作了大量的史料勾稽、考证工……

获取途径：

试读

版权页 | 前言页 | 目录页 | 正文页

文献传递：

图书馆文献传递

互助平台：文献互助

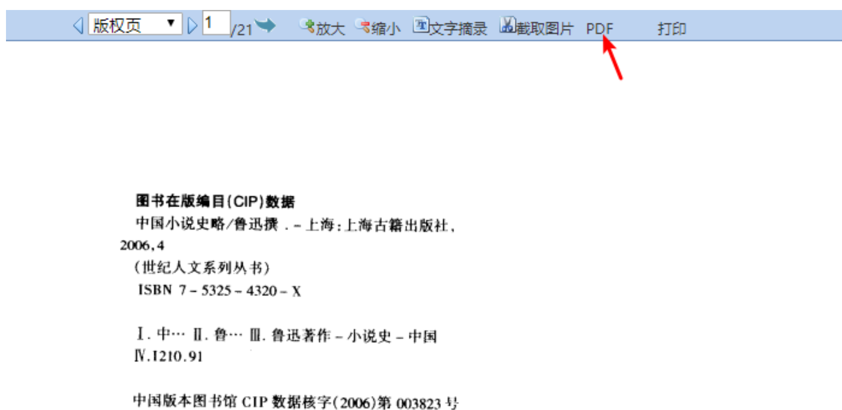
其他图书馆借阅

深圳之城纸书图书馆
福州大学图书馆
北京科技大学图书馆
深圳大学图书馆
国防科学技术大学图书馆
湖南科技大学图书馆
中南大学图书馆
昆明学院图书馆
湖南大学图书馆
湖南师范大学图书馆
更多...

图6-49　搜索结果详情页

下载辅助页

这会在心标签页中打开你所点击链接的预览页面，点击红色箭头所指的「PDF」链接：



图书在版编目(CIP)数据

中国小说史略/鲁迅撰．-上海:上海古籍出版社，2006,4

(世纪人文系列丛书)

ISBN 7-5325-4320-X

Ⅰ.中… Ⅱ.鲁… Ⅲ.鲁迅著作-小说史-中国 Ⅳ.I210.91

中国版本图书馆 CIP 数据核字(2006)第 003823 号

图6-50　预览试读页

这会在新标签页中打开一个页面，并让你选择下载正文试读页和辅助页。正文试读页一般为20页左右，辅助页一般包含封面、封底、扉页、版权页、前言/序、目录等页面。案例演示使用的浏览器是Chrome，我安装过PDF浏览程序，所以可以在浏览器中打开PDF文件。部分浏览器可能会在点击链接之后，弹出下载对话框，请选择保存的文件夹，然后保存。

点击「辅助页（版权页、前言页、目录页）下载」链接，在新标签页中打开PDF文件，点击红色箭头所指的下载按钮：



图6-51　预览及下载试读页

在弹出对话框中选择文件夹以保存改PDF文件。

提取封面

然后使用Adobe　Acrobat打开这个PDF文件，切换到「页面缩略图」模式，选择封面页，点击鼠标右键，选择「提取页面」选项：

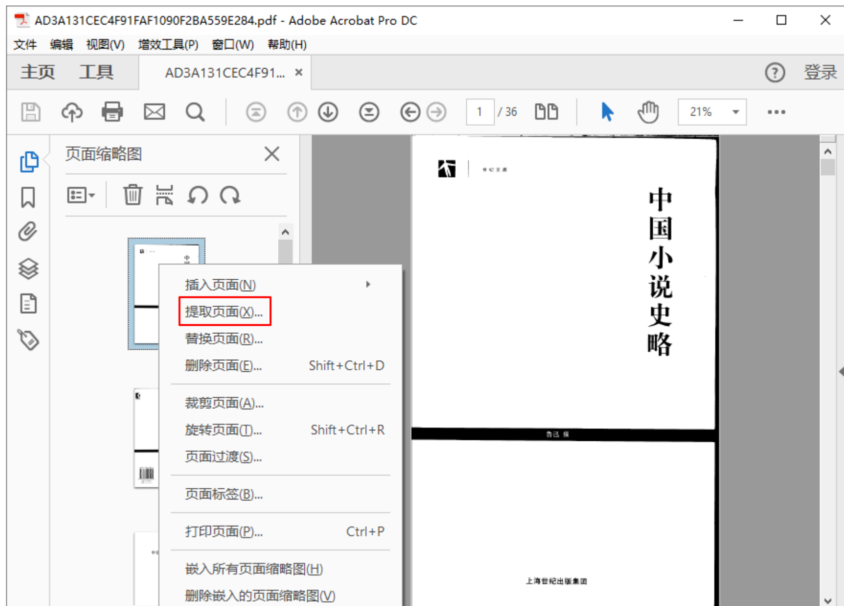


图6-52　提取页面

在弹出对话框中点击「确定」按钮，这会在Adobe　Acrobat软件的新标签页中打开提取的页面。依次点击「文件→导出到→图像→JPEG」：

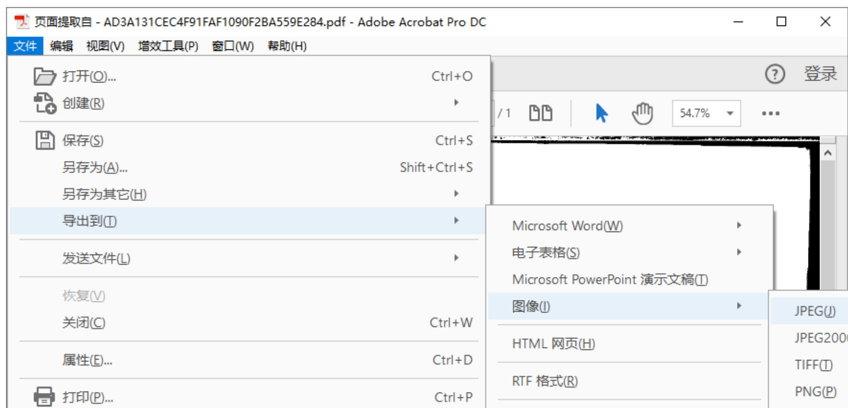


图6-53　导出提出的页面为图像

在弹出对话框中对图像进行重命名，然后点击「设置」按钮，在弹出窗口中设置如下：

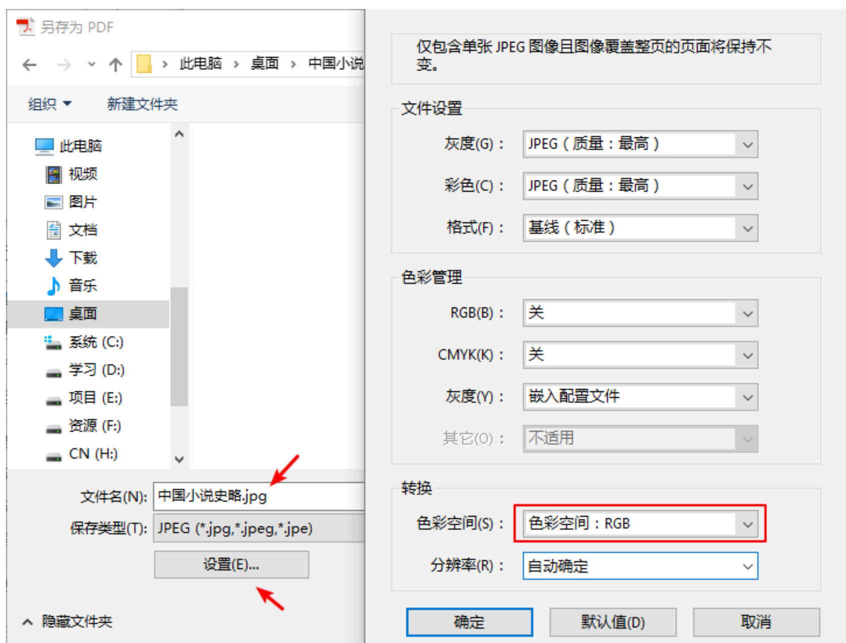


图6-54　设置图像的信息

然后点击「确定」按钮，在「另存为PDF」对话框中点击「保存」按钮。最后文件夹如下：



图6-55　提取的图像

封面文件如果有瑕疵可在Ps中进行修改。

6.3.2　立体封面修改为平面图片

立体封面一般是用来展示的，如果实在是找不到大的平面图片尺寸而又对从图书馆中下载的封面不满意的话，可以考虑用立体封面修改成平面图片。仍然以《中国小说史略》为例，这次换成立体封面。

在搜索引擎中搜索书名，挑选一张立体封面，如下：

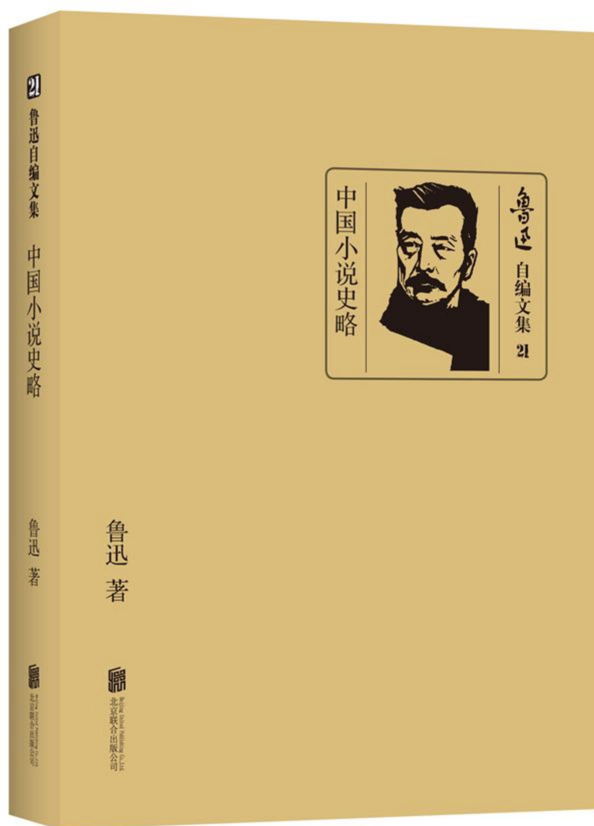


图6-56　待处理的素材

为了缩小所占空间，我为图像添加了宽度。你也可以把这张图片另存到你的计算机中，跟着教程一步步练习。首先启动Photoshop软件，打开这个封面图片：

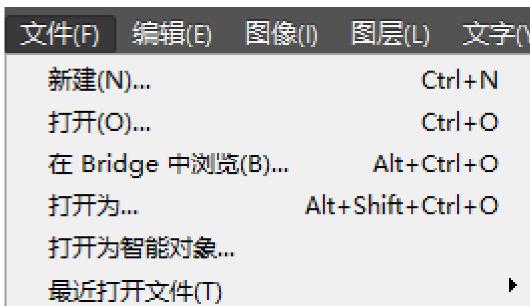


图6-57　打开待处理的图片

依次点击「文件→打开」，在打开文件对话框中找到你保存封面的文件夹，然后选择并打开它，或者使用快捷键「Ctrl＋O」。此时Ps界面如下：

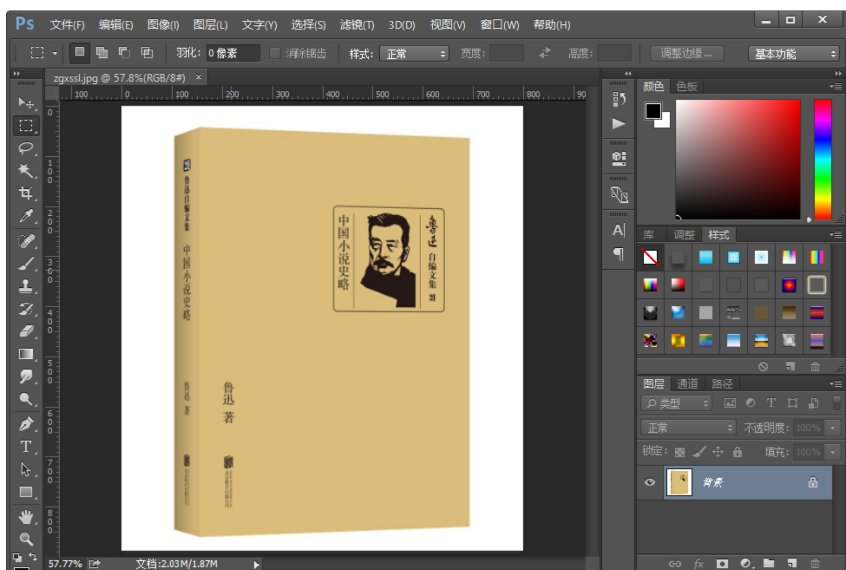


图6-58　缩放图像

你可以选择放大软件操作界面为全屏，然后按快捷键「Ctrl＋0」，按屏幕大小缩放图片。或者依次点击「视图→按屏幕大小缩放」。



图6-59　按屏幕大小缩放图像

这时封面图片会以当前编辑界面的高度显示。你也可以按快捷键「Ctrl＋1」，以100%的比例显示封面图片。

如果你的标尺工具没有显示出来，请使用快捷键「Ctrl＋R」显示你的标尺。标尺即上上图中的编辑界面顶部和左侧的刻度。鼠标放置在标尺上，点击鼠标左键，向下或向右拖出参考线，完成后如下：

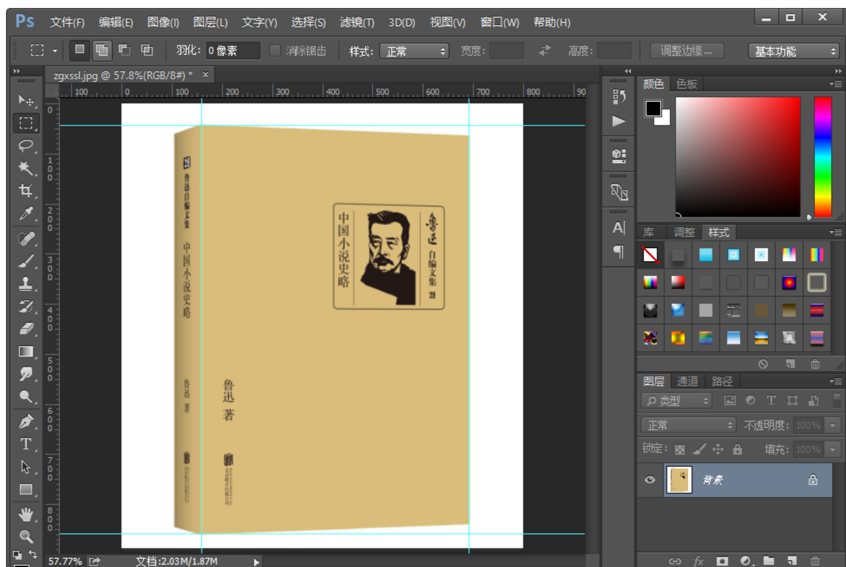


图6-60　设置参考线

在左侧的工具栏按钮中选择第2个工具——矩形选框工具。鼠标移动到参考线所画的矩形中，点击鼠标左键画矩形虚线框，然后按住Space键（空格键）可以移动这个虚线框。当虚线框移动到参考线附近时，会有

一个吸附效果。

把虚线框移动到矩形的左上角，贴近参考线时，放开按住空格键的手，这个期间不要放开按住鼠标左键的手指。扩展这个虚线框，使之与参考线所画的矩形完全重合。完成后如下：

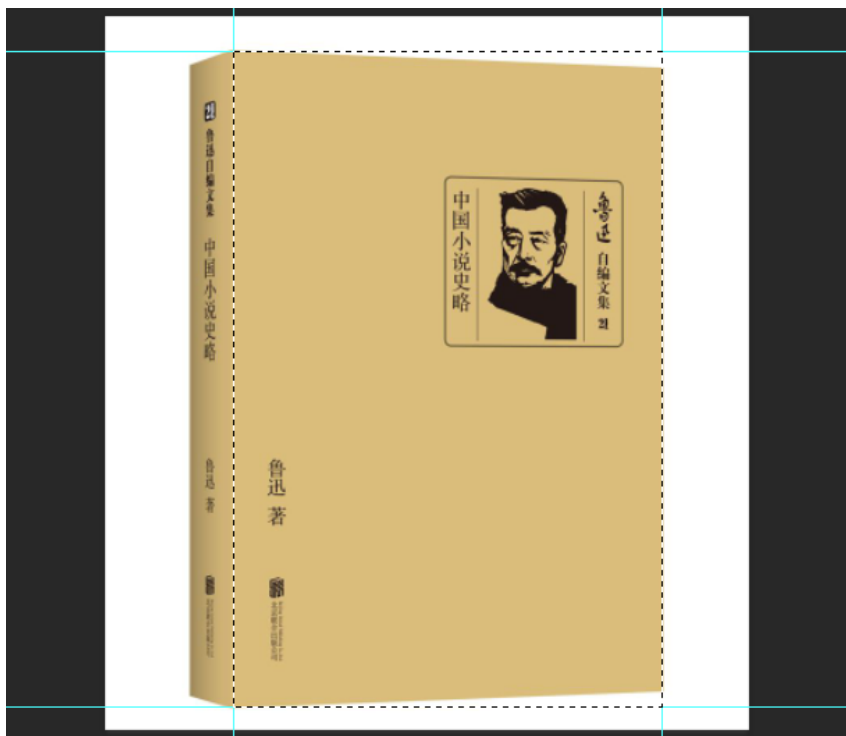


图6-61　建立选区

按快捷键「Ctrl+T」进入「自由变换」模式，点击鼠标右键选择「扭曲」：



图6-62　扭曲选区

鼠标移动到虚线框的右下角，此时鼠标变成三角形状，点击鼠标左键向下移动，同时按住Shift键。使封面图底部与参考线重叠，完成后如下：

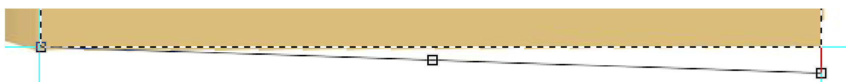


图6-63　调整选区

封面右上角同样如此操作，区别是向上移动。完成后点击Enter键（回车键），退出「自由变换」模式。按住快捷键「Ctrl＋D」撤销虚线框选区。

再重新画虚线框，框选参考线所画的矩形。关闭输入法或切换到英文状态，按下C键，进入裁剪模式。或者点击左侧工具栏中的第5个工具按钮——裁剪工具。

按下Enter键（回车键）确认裁剪，然后点击状态栏中的「√」按钮，或者再次按下Enter键（回车键）。完成后会把选区之外的区域删除，只保留虚线框中的内容。如下：

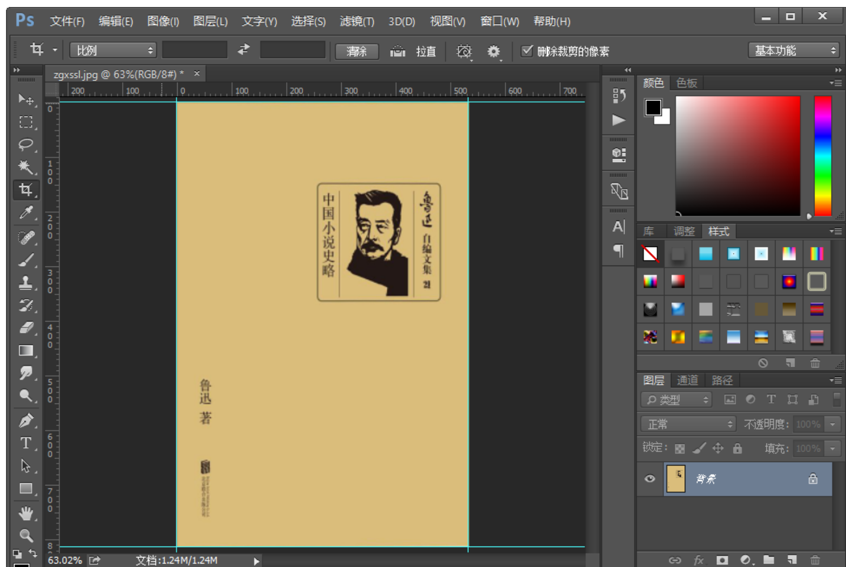


图6-64　裁剪选区

到了这一步，基本上封面已经出来了。但是，我们仔细观察，会发现，封面的宽高比有一点不协调。这种情况执行如下操作：使用「矩形选框工具」框选整个封面，按下快捷键「Ctrl＋T」进入「自由变换」模式，鼠标移动到封面上部红框区域：

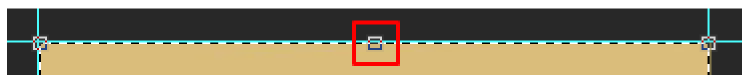


图6-65　调整图像比例

此时鼠标变成可以上下拉伸的状态，因为我们要调整宽高比，所以选择向下拖拽。向下拖动，调整到一个合适的范围。按下C键，再按2次Enter键（回车键），裁剪封面。

裁剪完之后，按快捷键「Ctrl＋S」保存封面，或者依次点击「文件→存储」。



图6-66　存储修改的图像

在弹出对话框中选择文件夹，然后保存封面，可根据需要对封面文件进行重命名。

Photoshop的「裁剪工具」中有一个「透视裁剪工具」也可以实现图6-64的效果，实现过程比建立参考线的方式要快很多。

6.3.3 封面重制

在Ps中查看上一节处理的封面，宽度533像素、高度747像素，远没有达到最基本的720×1280的要求。仍然以上一节的封面为例，进行封面重制。在重制一本书的封面之前，我们先观察封面，确定哪些元素是要用到网络中的素材，哪些元素是可以在Ps中绘制的。对于需要用的素材，可以在网络中收集。尽量使用高清的素材。

我们来观察下《中国小说史略》这本书的封面：

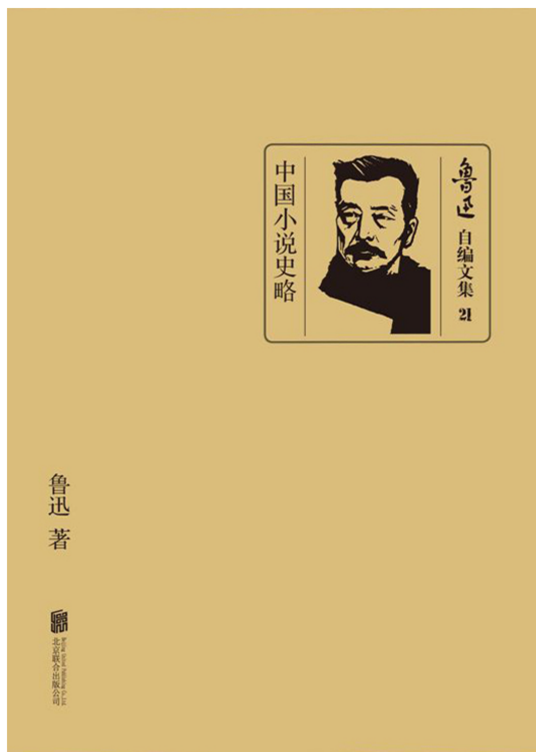


图6-67　《中国小说史略》封面

从上图中可以看到，除了鲁迅头像、出版社标、鲁迅二字之外，其他元素都可以在Ps中做出来。

字体方面，我们可以使用免费的思源宋体和思源黑体。字体下载完成之后，点击鼠标右键选择「安装」即可。

图片方面我们可以使用以图搜图，出版社标可以通过关键字搜索，鲁迅签名同样。素材收集如下：

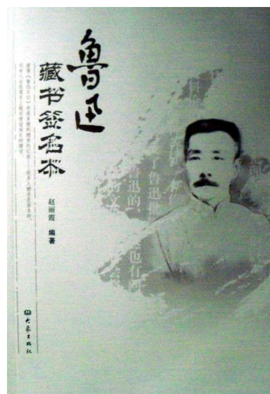





图6-68　制作封面的素材

首先处理「鲁迅」二字，用Ps打开这张图片。使用「矩形选框工具」框选签名，然后裁剪页面。完成后如下：



图6-69　解锁图层

在右侧的图层中，点击锁定按钮，解锁图片。

按下快捷键「Shift＋Ctrl＋U」去除图片颜色，或者依次点击「图像→调整→去色」：



图6-70 图片去色

按下快捷键「Ctrl＋L」调出「色阶」设置窗口，或依次点击「图像→调整→色阶」，移动「输入色阶」中的滑块，使素材完全变成黑白二色：

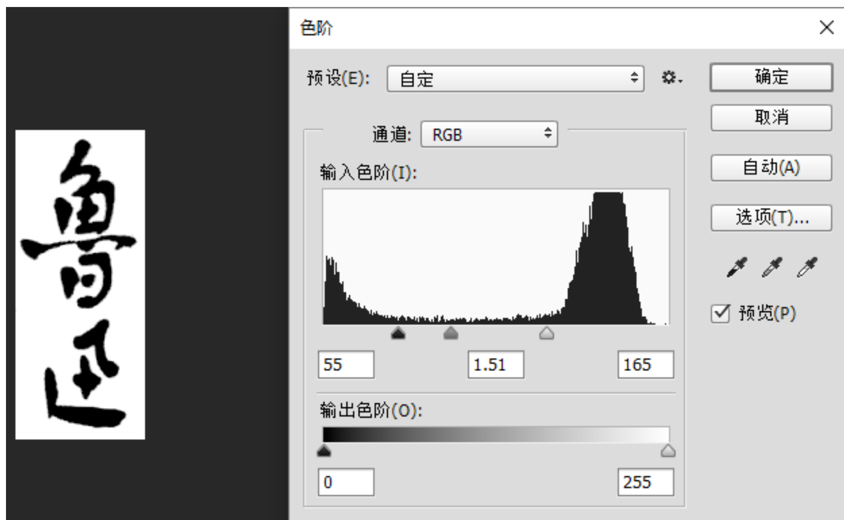


图6-71　调整色阶

点击「确定」按钮，退出编辑。点击左侧工具栏中的第4个工具按钮——魔棒工具。此时鼠标变成一个魔棒形状，在黑色部分点击鼠标左键，或者点击白色区域，然后按下快捷键「Shift＋Ctrl＋I」。完成后如下：

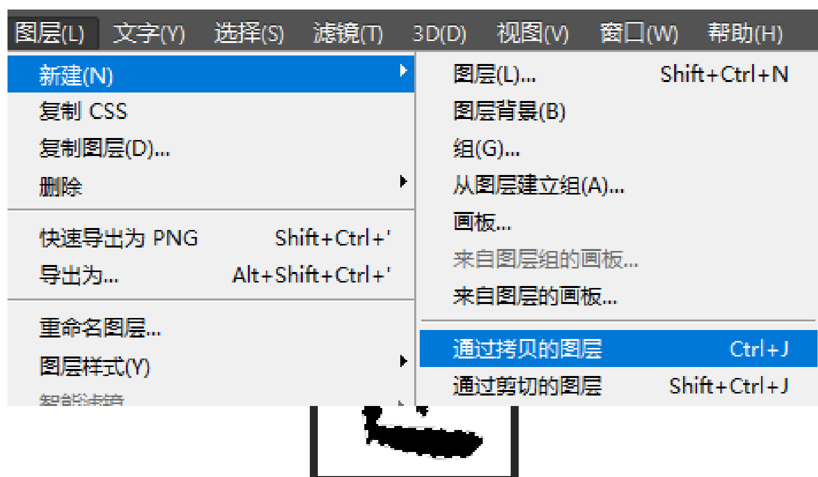


图6-72　建立选区

按下快捷键「Ctrl＋J」复制到新图层中，或依次点击「图层→新建→通过拷贝的图层」。

这是我们可以先隐藏原来的图层，只保留刚刚复制的。以PSD文件格式保存。

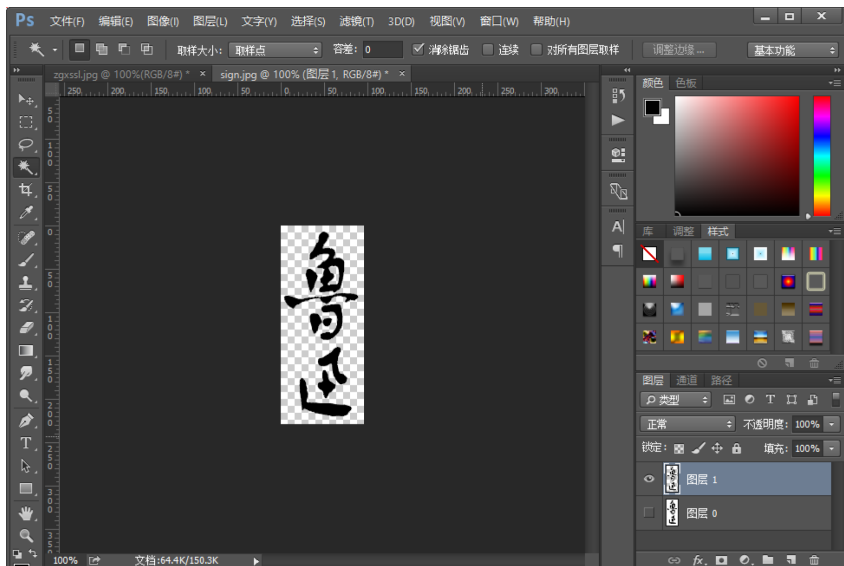


图6-73　删除背景

然后是出版社标，这张图片处理起来比较简单，只需要去除颜色就行，步骤参考上文中的步骤。

接着开始处理鲁迅头像，仔细观察本节开始的封面，我们发现素材与最后成品的区别并不是特别大。这里我们使用橡皮擦工具。打开头像图片，点击Ps左侧的工具栏的第11个工具——橡皮擦工具。

你可以输入法关闭或英文状态下使用[]键放大或缩小橡皮擦工具，在涂抹的过程中可以对图片进行放大。缩放工具是Alt键配合鼠标滚轮。步骤：按住Alt键，向前或向后滚动滚轮。

当图片放大到一定程度时，图片会覆盖整个编辑界面，按住Space键（空格键）+鼠标左键，在编辑界面移动图片。完成如下：

图6-74　缩放图像大小

参考上一节，把人像抠出来。以PSD文件格式保存。

然后根据本章开始的介绍语新建一个文件，高2048×宽1536。按快捷键「Ctrl+N」或依次点击「文件→新建」：

文件(F)	编辑(E)	图像(I)	图层(L)	文字(Y)
新建(N)...	Ctrl+N			
打开(O)...	Ctrl+O			
在 Bridge 中浏览(B)...	Alt+Ctrl+O			
打开为...	Alt+Shift+Ctrl+O			
打开为智能对象...				
最近打开文件(T)				

图6-75　新建文件

在弹出设置窗口中设置如下：



图6-76　填写新文件参数

然后点击「确定」按钮。

打开上一节处理好的封面图片，依次点击「窗口→排列→双联垂直」，这一步是为了我们在重制封面时有原图在旁边可以参考细节：



图6-77　双联垂直摆放已打开窗口

然后为新建的文件设置背景色，首先选择背景图层，「Ctrl＋J」复制一个新图层，然后点击左侧工具栏中的前景色图标。

在弹出的「拾色器」窗口中，鼠标移出这个窗口外会变成一个吸管形状，点击原来的封面图片，拾取颜色：

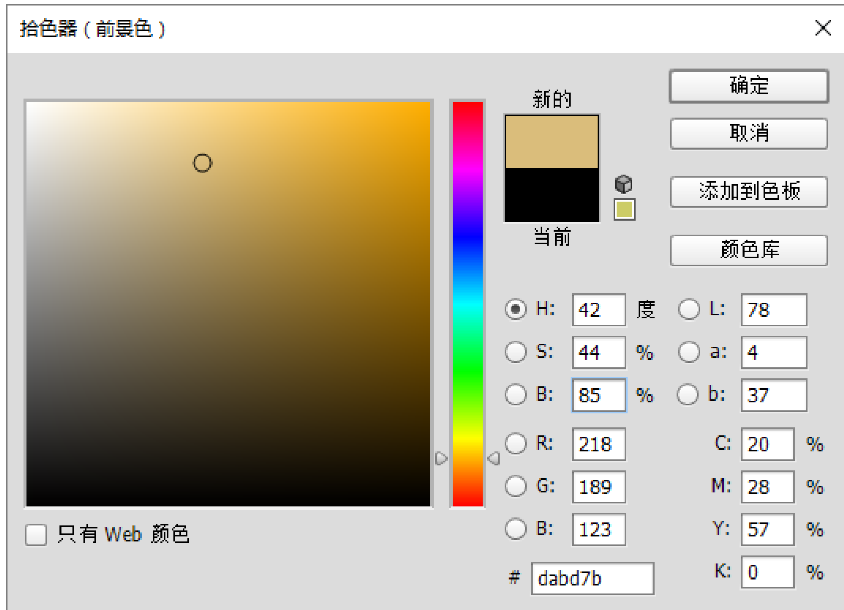


图6-78　拾取颜色

然后点击「确定」按钮，此时前景色图标为 。然后选择新建的文件标签页，按快捷键「Alt＋Delete」填充前景色。完成后如下：

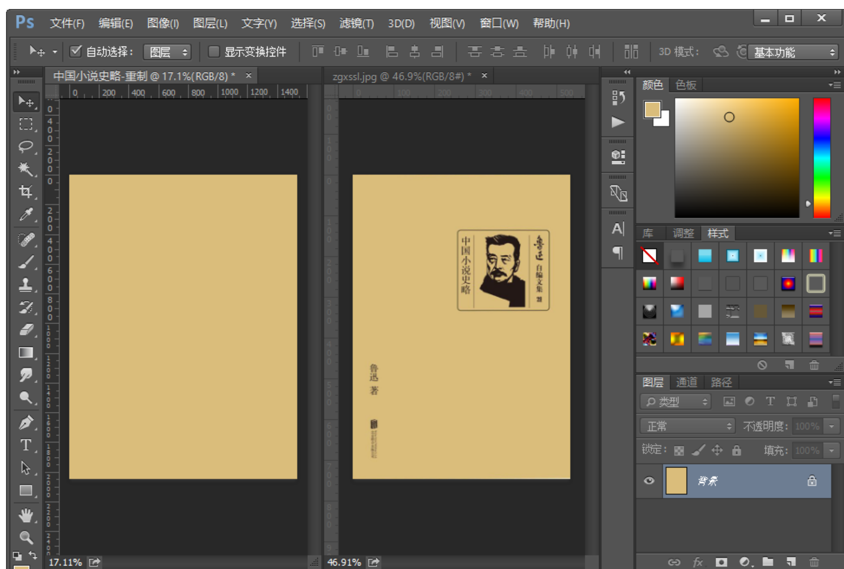


图6-79　填充背景色

剩下的就是按照原封面的位置摆放素材了。

使用文字工具 输入文本，输入完成后可以在左上角调成竖排形式，

并调出字符设置面板，设置字体、字号等：



图6-80　字符设置面板

所有按钮在文字工具中的状态面板中：



图6-81　字符工具栏

字符面板按钮是倒数第2个图标，或依次点击「窗口→字符」。

打开出版社标的PSD文件，选择是透明的图层，按住Ctrl键点击图层缩略图，会自动为图片创建选区。然后选择左侧的工具栏第1个工具——移动工具。鼠标移动到选区上，然后按住鼠标左键移动图像到新建封面的标签标题上，标签页会进行切换。松手就行，需要再移动图像时，先选择图层，再移动图像。

如果图像过大或过小，使用「Ctrl＋T」快捷键，进行缩放。缩放时，鼠标在图像的节点上，比如图像的4个角。缩放时按住Shift键或Alt键会有不同的缩放效果。

按住Ctrl键点击图层缩略图，然后再切换到要对齐的对象，再选择「移动工具」，在状态栏中有水平、垂直等对齐方式可以选择。然后对齐刚刚输入的文本。

按Shift键＋←→键可以向左或向右一次移动10像素。

接着输入出版社中文名和英文名，转成竖排、更换字体、改变字号等。

多个图层可以先选择，然后使用快捷键「Ctrl＋G」进行编组，编组之后的组内图层可以使用「自由变换」进行放大或缩小。完成后如下：



图6-82　素材摆放

每一步完成后记得及时保存！

文字工具的下下方，有一个「圆角矩形工具」，选择之后，画一个带有描边的圆角矩形。

在「圆角矩形工具」上方有一个「路径选择工具」，在状态栏中设置如下（填充无，描边5像素），描边可能默认单位是「点」，点击鼠标右键可以从右键菜单中选择「像素」：



图6-83　修改路径状态

然后依次点击「窗口→属性」，调出「属性」面板，设置该矩形框圆角半径为20像素：



图6-84　形状属性设置面板

然后输入文字，摆放素材。矩形框中的竖线，长按「圆角矩形工具」后，选择「直线工具」。在状态栏中设置线条粗线，然后按住Shift键画直线。完成后如下：

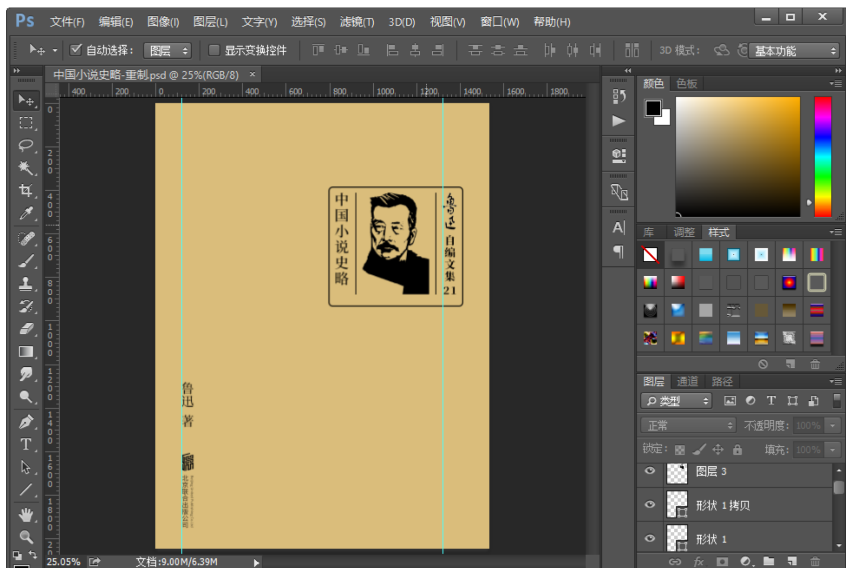


图6-85　素材排版

完成后可以进行保存操作，也可以另存为JPG格式。

6.3.4　修改封面尺寸

在本章开始部分讲过要对平板和手机进行封面适配，所以需要制作两种不同尺寸的封面。

打开上一节制作的高清封面（JPG格式），按「Alt+Ctrl+I」快捷键或依次点击「图像→图像大小」，弹出「图像大小」设置窗口：

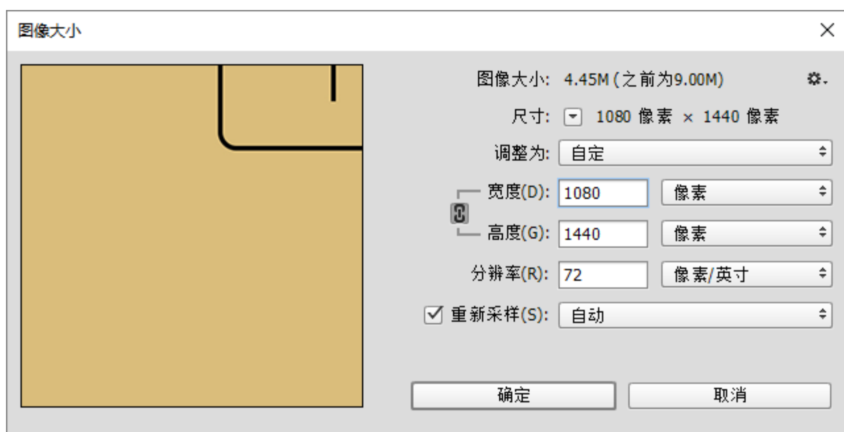


图6-86　「图像大小」设置窗口

输入宽度1080、分辨率72，单位像素。参数如上。

然后点击「确定」按钮，按「Alt+Ctrl+C」快捷键或依次点击「图像→画布大小」，弹出「画布大小」设置窗口：

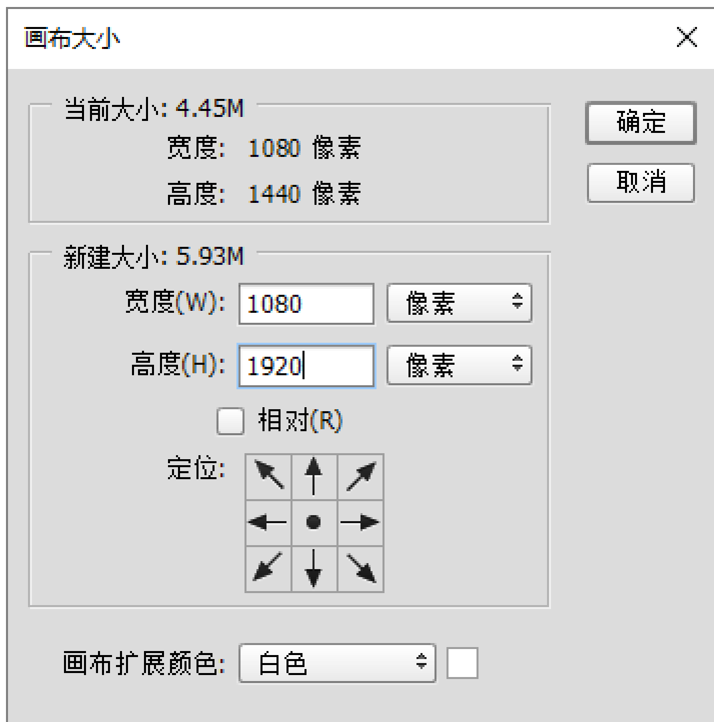


图6-87　「画布大小」设置窗口

输入高度1920，单位像素。参考如上。完成后如下所示：



图6-88　上下扩展画布

从图中可以发现，高度进行了上下补白（「画布大小」中设置的画布扩

展颜色为「白色」)。然后使用「矩形选框工具」选择白色区域:



图6-89　填充背景

然后在虚线框中,点击鼠标右键,选择「填充」或按「Shift+F5」快捷键,调出「填充」窗口:



图6-90　填充选项

点击「内容」下拉框,选择「颜色」,在弹出的「拾色器」窗口中拾取背景颜色。点击「确定」、「确定」按钮。封面底部的白色区域,同样如此操作。

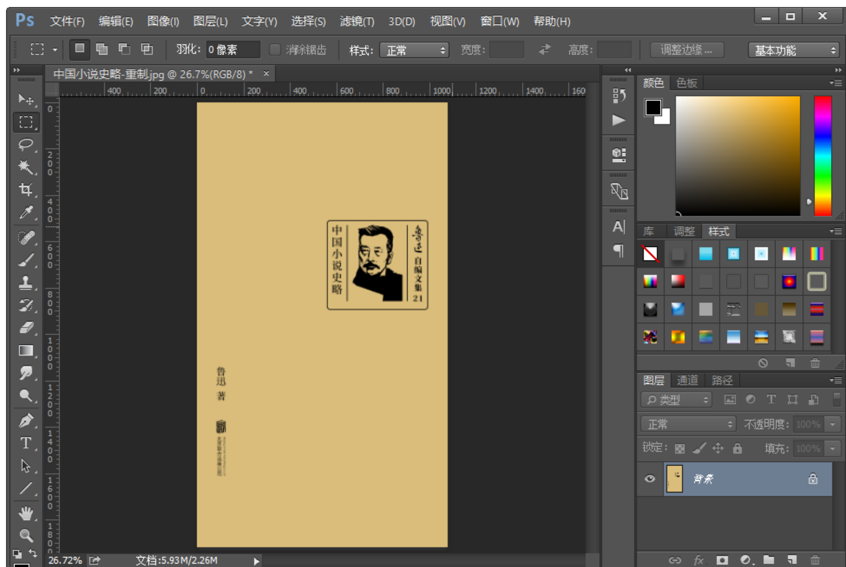


图6-91　修改好的手机版封面

如果背景都是这种纯色的封面，可以使用这个操作。修改完成后保存文件，为与平板封面区别，可另外命名。

如果封面背景是图案的话，「填充」的内容则是「内容识别」。背景图案特别复杂的话，「内容识别」可能也不会好用。可以适当地拉伸图片，使图片符合手机封面的要求。

> 封面上的主要信息一定要在显示设备的可视区域内，避免不同尺寸的设备裁切封面，把封面信息裁剪掉。

6.4　生僻字图片制作

有些生僻字，使用输入法打不出来，还有的字，限于阅读系统的字符编码，显示不出来。这种情况，我们就需要把生僻字做成图片，来达到显示的目的。

整体思路是把一个生僻字拆分为不同的部首，把部首提取出来，再合成一个字。生僻字图片做出来后，扣掉底色，设置高度或宽度为一个汉字高度。

对于阅读系统不支持显示的字型，在电脑上安装一个超大字符集，在PS里打出来。对于在超大字符集也没有的字型，则需要找一个字型相近的字，替换部首，完成造字。以下是具体实例：

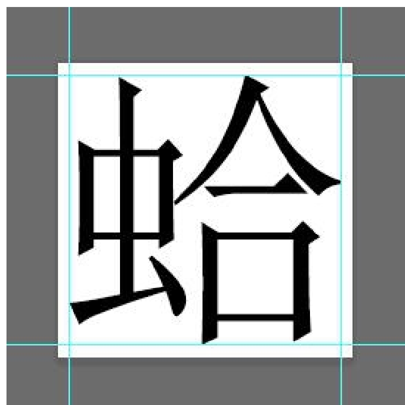
比如我想造一个汉语大字典中的字，输入法打不出来。截图：

蛪 guǎi

蛙类。种类甚多，如山蚂蛪，犁头蛪等。清李調元《南越筆記》卷十一："蛪者，蛤之屬。諺曰：'蟾蜍、蛤、蛪。'三者形狀相似，而廣州人惟食蛤，不食蟾蜍、蛪。"

图6-92　需要造的字型

> 观察字体结构，这个字是左右各一半——虫和另，比较好找字型相近的字来合并。这里用「蛤」和「别」为例。



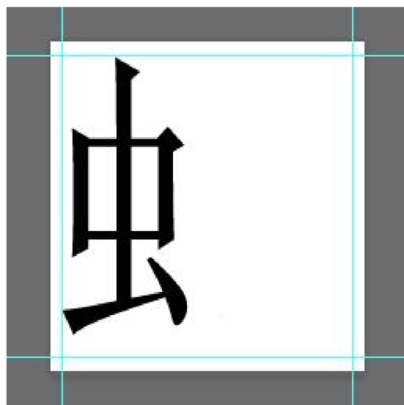


图6-93　取「蛤」字偏旁「虫」

1.在PS中新建画布128px×128px（保证即使放大，也不损失清晰度），打出第一个字：蛤，调整字体大小，一般为画布的宽度值，水平垂直居中这个画布。注意：如果书中是楷体，就把字体设置为楷体，其他情况类推，这里以宋体做为演示；

2.选中文字图层，右键执行「栅格化文字」操作，用「橡皮擦」工具擦掉左边的「合」旁，然后隐藏该图层；

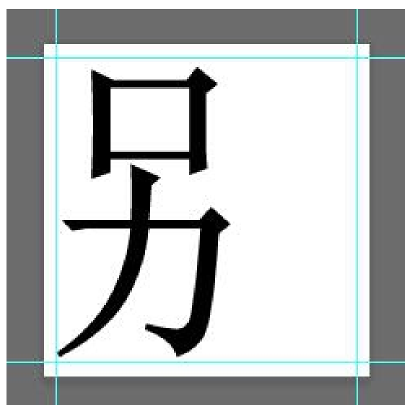




图6-94　取「别」字偏旁「另」

3.先隐藏这个文字图层，再打出第二个字「别」。设置大小，调整字体，水平垂直居中，再进行「栅格化文字」操作。用橡皮擦擦掉「刂」旁；

4.移动「别」文字图层至左边，紧挨参考线，显示「蛤」文字图层：

5.现在就把字造好了。做好的生僻字需要调整间距，这个还是要用肉眼判断下，调整到舒适即可。举例的这个字，比较简单，所以基本不用调整。建议善用参考线。

6.先隐藏背景图层，另存为一个背景透明的png格式图片文件。然后是

图片添加到到ePub的Images文件夹下中（方法见5.3.13节）。插入到段落中，代码如下：

HTML代码：

```
<img class="rarefont" alt="生僻字" src="../Images/
rarefont_1.png"/>
```

CSS代码：

```
img.rarefont {
    margin: 0; /*清除外边距*/
    padding: 0; /*清除内边距*/
    height: 1em; /*高度为一个汉字的高度*/
    vertical-align: middle; /*垂直居中*/
}
```

在手机中的预览效果（多看阅读）：

广西的河池、百色及凤山、东兰一带的壮族，在农历正月期间，要过一个与农事活动紧密有关的“蚂蜗〔guǎi拐〕节”（青蛙节）。据说古代这里壮族有一个名叫东林的孝子，他的母亲去世时，屋外蛙声不绝，他怕打扰了母

23:21 69 / 235

图6-95　行内字预览

这样就完成了一个生僻字图片的制作。可以比较下经过字体设计师和自己用PS制作的不同。

图6-96　字体设计师设计的

图6-97　Ps软件合成的

关于插入行内图片的更多细节，见10.7节。

第7章

文本处理——EmEditor

EmEditor是江村软件公司（Emurasoft，坐落于美国华盛顿州）所开发的一款在Windows平台上运行的文字编辑程序。EmEditor以运作轻巧、敏捷而又功能强大、丰富著称而得到许多用户的好评。Windows内置的记事本程序由于功能太过单薄，所以有不少高级用户直接以EmEditor等第三方文字编辑程序取代。

EmEditor支持多种配置，自定义颜色、字体、工具栏、快捷键设置，可以调整行距，避免中文排列过于紧密，并允许无限撤消、重做，功能丰富多样，使用方便，是替代记事本的最佳编辑器。

7.1 常规设置

演示所用的版本为V17.1.0版本，可自行搜索安装包，这里不提供。

软件在使用前可以按照下面的教程进行配置，这样可以一边熟悉软件，一边提升一部分工作效率。

依次点击「工具→当前配置属性」，选择左侧的「常规」：

图7-1 Text属性-常规

「自动换行」点击下拉按钮，选择「窗口宽度」，勾选右侧的「显示行号」。显示行号后，在EmEditor中查看打开的文件可以显示每段的行号。

然后点击右下角的「制表符/缩进」按钮：

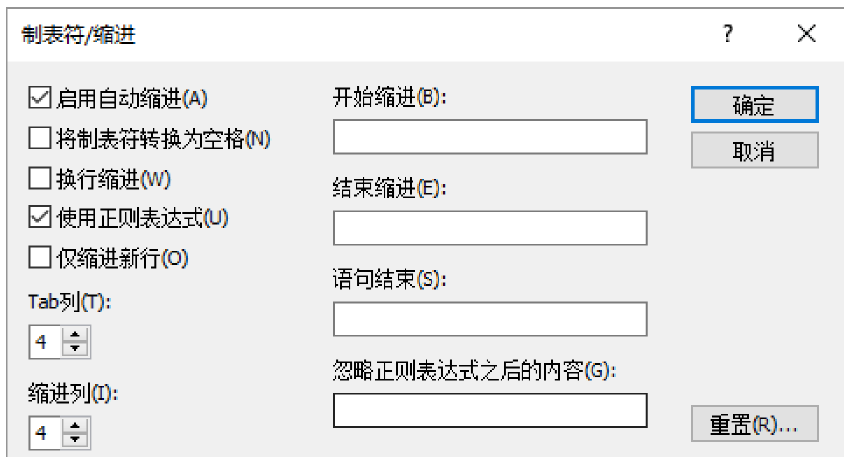


图7-2　制表符/缩进设置

勾选「启用自动缩进」和「使用正则表达式」，Tab列设置为4，缩进列为4。启用自动缩进后，在一段话结束换行后，会自动添加缩进字符，如全角空格。

选择左侧的「滚动」，勾选「滚动选项」中的「总是启用1页垂直滚动」。

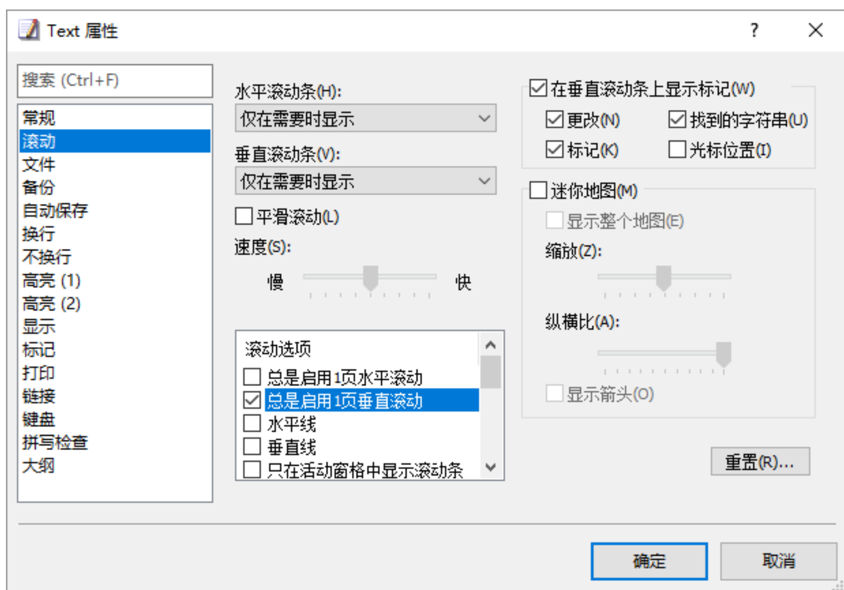


图7-3　设置1页垂直滚动

在使用编辑器的过程中，您肯定遇到过这种情况：编辑到了文件末尾，看着不方便，然后你就会按很多次回车来产生很多空行，以使编辑行在中间。EmEditor提供了很好的解决方案，只需要勾选上「总是启用1页

垂直滚动」，EmEditor便一直有一空页，让你的编辑总是在上面或中间，而且没有产生多余的空行。

「滚动选项」中的「水平格线」勾选后如下：

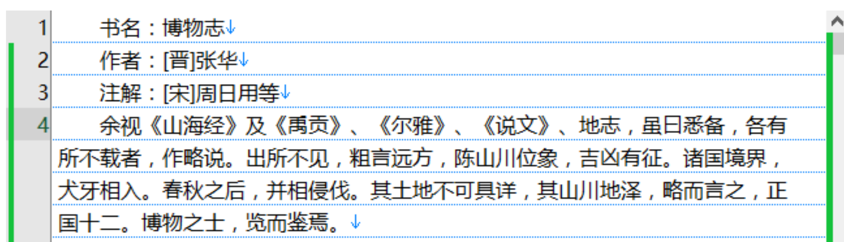


图7-4　水平格线

如果您喜欢可以勾选上。「水平线」和「垂直线」你可以勾选上，当你在不同的位置插入光表示，会有一纵一横两条线在光标位置交叉，提示你光标此时所在的位置。

选择左侧的「自动保存」，勾选「自动保存间隔」，输入1：

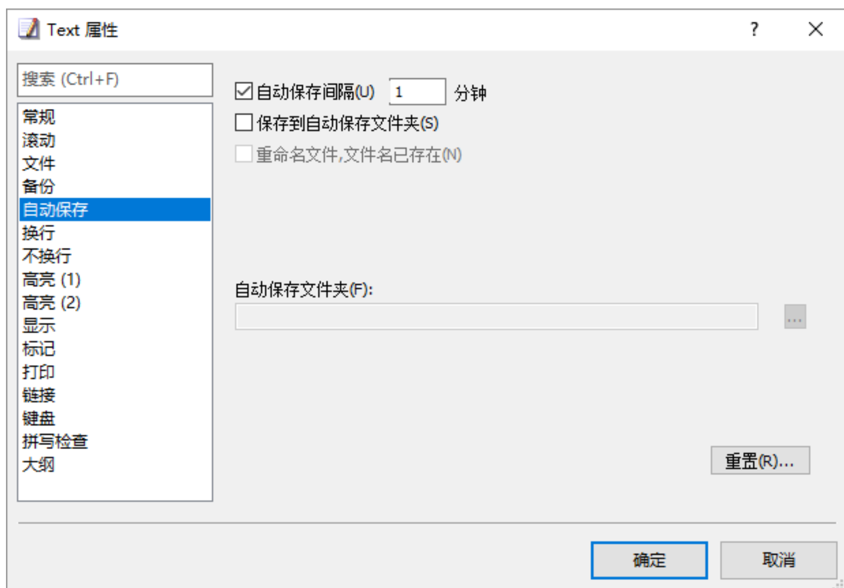


图7-5　自动保存设置

虽然部分软件可以设置「自动保存」，但是强烈建议你在编辑之后养成随手按Ctrl＋S（保存）的习惯。

选择左侧的「高亮(2)」勾选「高亮配对括号」和「自动完成括号/引号标记」，「自动完成括号/引号标记」勾选后，输入左侧括号，比如输入【符号，软件会自动补上】，并把光标插入到括号之间：

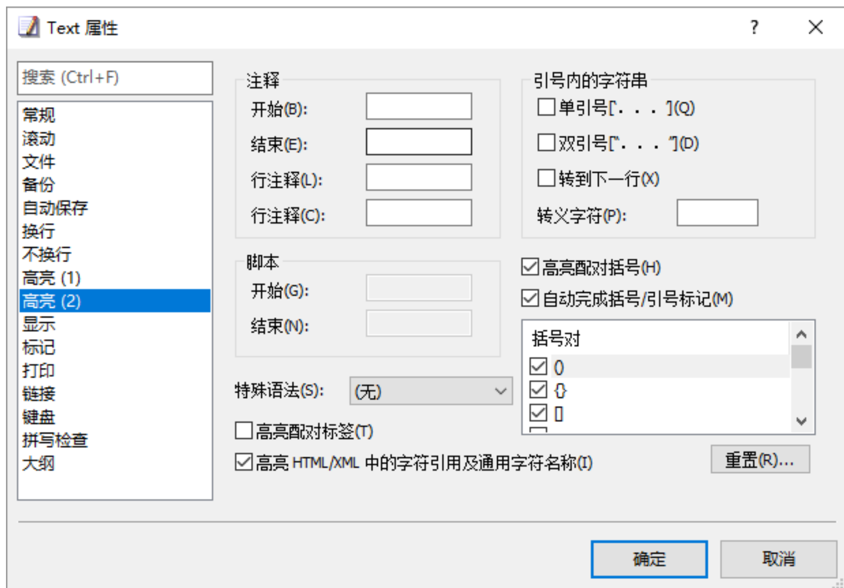


图7-6　特殊文本高亮

选择左侧的「显示」，可以从默认主题中选择，然后根据个人习惯进行修改（下文中有配置Text的教程，可参看）。

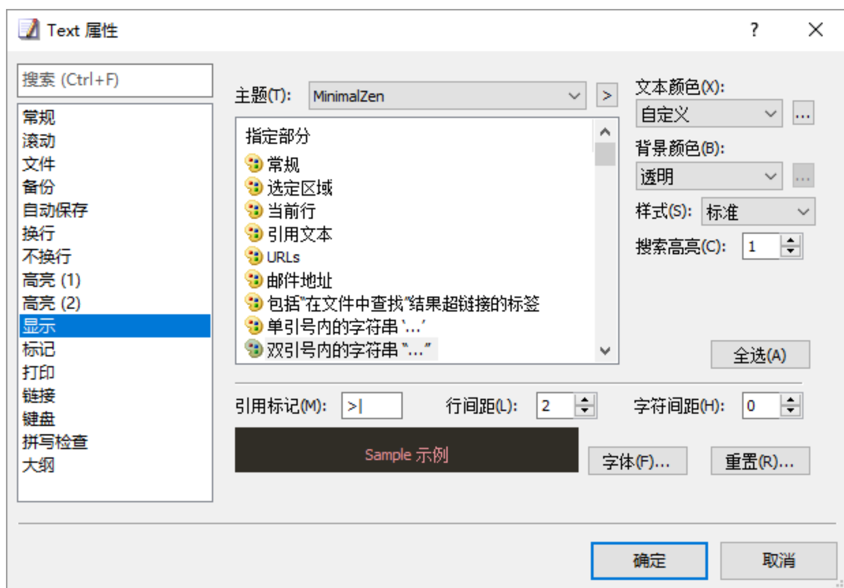


图7-7　设置行间距

其中「行间距」可以调整行高，如果你觉得中文排列比较紧密，可以增大行高。

选择左侧的「标记」勾选「显示空格」和「显示全角空格」：

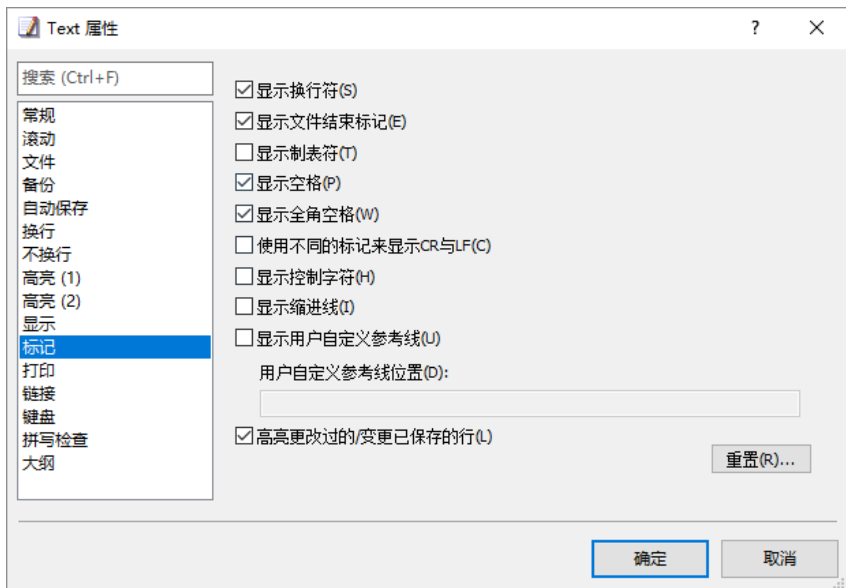


图7-8　全角空格与半角空格标记

「显示换行符」勾选后会显示为↓，和「显示文件结束标记」一起为默认勾选，建议保留。

配置完成后，点击右下角的「确定」按钮，保存配置。

7.2 配置Text

由于我们主要是处理TXT文本的，所以可以单独针对Text做一些个性化配置。配置好的显示如下：

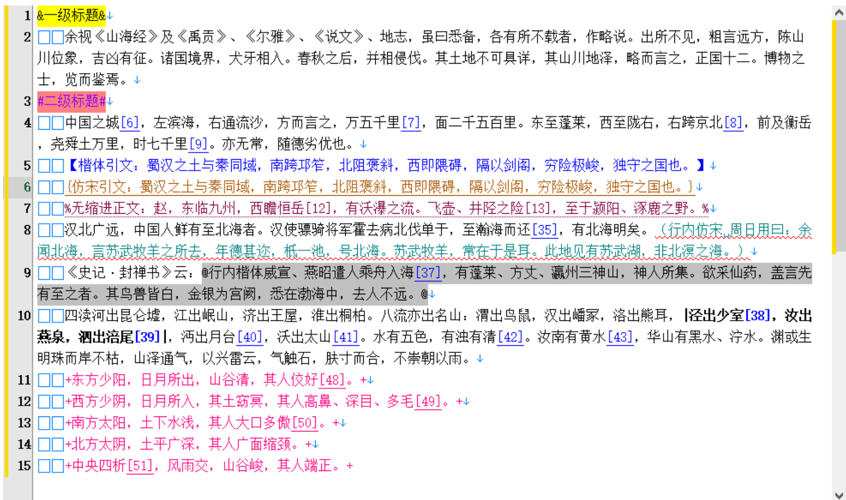


图7-9　配置好的Text属性

样式列表：

一级标题：&内容&
二级标题：#内容#
注释序号：[数字]
楷体引文：【内容】
仿宋引文：{内容}
无缩进正文：%内容%
行内仿宋：（内容）
行内楷体：@内容@
行内加粗：|内容|
诗歌居左：+内容+

EmEditor只能配置10个高亮。复制下面的代码，然后新建一个文本文档，粘贴代码，修改文件类型为「.esy」，重命名然后保存：

```
; EmEditor Syntax File
;
; To import this file to an existing configuration, select Configuration
; under the Tool menu, select Define Configurations, select a configuration
; you want to import to, press [Properties], select the Highlight page, and
; press [Import].
;
; Submit your customized file to submit@emurasoft.com
; to be listed in www.emurasoft.com user files pages.
;
#Highlight=on
#BeginTag=
#EndTag=
#CommentBegin=
#CommentEnd=
#LineComment1=
#LineComment2=
#SingleQuote=off
#DoubleQuote=off
#ContinueQuote=off
#Escape=
#ScriptBegin=
#ScriptEnd=
#SpecialSyntax=off
#HighlightBraces=on
#Keyword color=1,word=off,rightall=off,case=off,insidetag=off,regex
&.*?&
#Keyword color=2,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
^#.*?^#
#Keyword color=3,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
\[\d+\]
#Keyword color=4,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
【.*?】
#Keyword color=5,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
{.*?}
#Keyword color=6,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
%.*?%
```

```
#Keyword color=7,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
(.*?)
#Keyword color=8,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
\+.*?\+
#Keyword color=9,word=off,rightall=off,case=off,insidetag=off,regexp=on,righ
\|.*?\|
#Keyword color=10,word=off,rightall=off,case=off,insidetag=off,regexp=on,rig
@.*?@
```

依次点击「工具→当前配置属性」，选择左侧的「高亮(1)」，设置如下图：

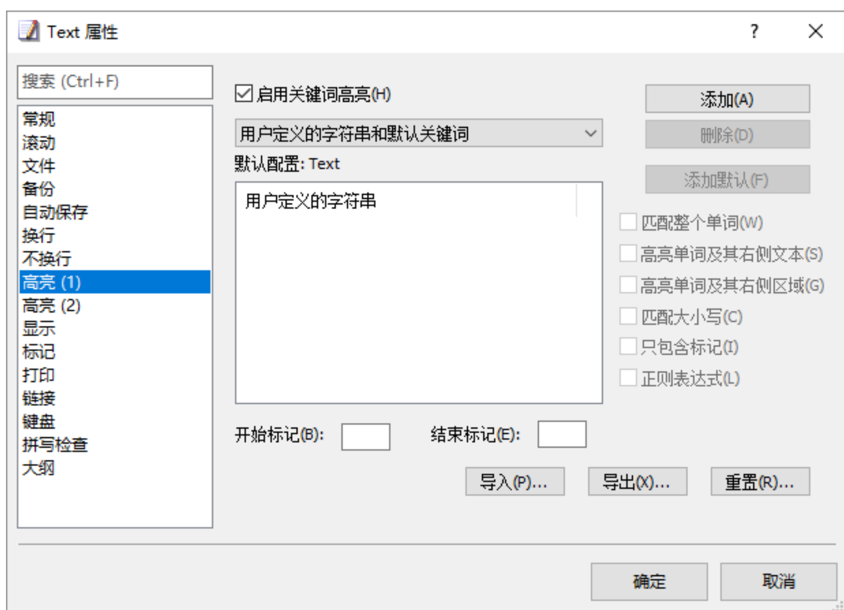


图7-10　自定义字符串高亮

点击底部的「导入」按钮，在弹出对话框中选择刚修改的esy配置文件，打开它。完成后如下图：

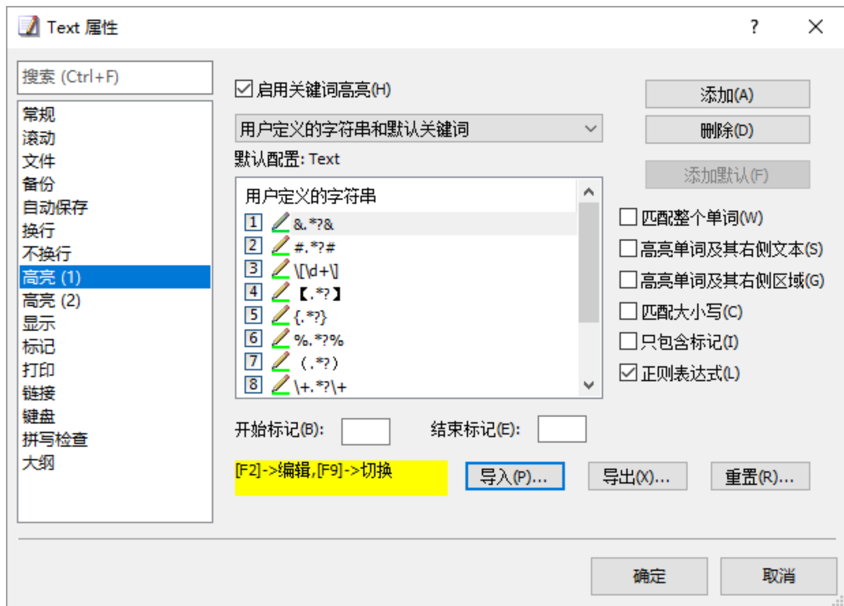


图7-11　自定义字符串高亮

然后点击右下角的「确定」按钮，保存配置。

配置生效后可能与上文截图不一样，你可以选择左侧的「显示」菜单，设置高亮的样式：

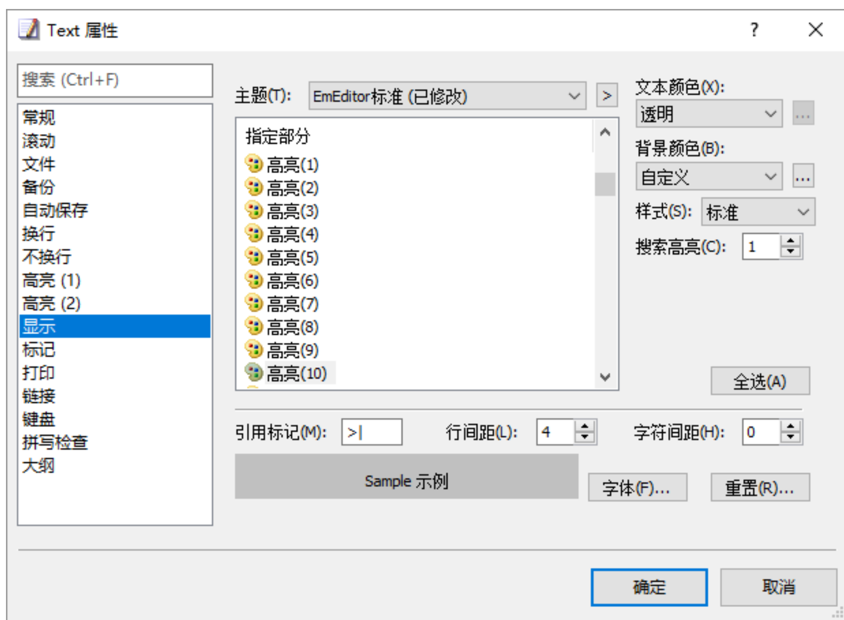


图7-12　自定义字符串高亮

上面的内容只是作为一个示例，提供方法，实际上你可以子集设计一套整理标准，或者采用通用的标准，比如markdown语法。

7.3　批处理

在EmEditor中可以使用字面值、正则表达式的查找替换。其中某些字符集的写法与Sigil不同，如查找中文字符：

```
Sigil：([\x{4e00}-\x{9fa5}])
EmEditor：([\u4e00-\u9fa5])
```

其他正则的书写与Sigil基本都差不多。

但是我使用EmEditor最多的功能是批量处理的功能，即预先设定好查找替换的内容，软件自动进行每个条目的查找替换。下面以一篇故事为例：

> &燕昭王
>
> 燕昭王者，噲王之子也。及即位，好神仙之道。仙人甘需臣事之，爲王述昆臺登真之事，去嗜欲，撤聲色，無思無爲，可以致道。王行之既久，谷將子乘虛而集，告於王曰：「西王母將降，觀爾之所修，示爾以靈玄之要。」後一年，王母果至。與王遊燧林之下，説炎皇鑽火之術，然绿桂膏以照夜〔一〕。忽有飛蛾銜火，集王之宫，得圓丘砂珠〔二〕，結而爲佩〔三〕。王登捱日之臺〔四〕，得神鳥所銜洞光之珠，以消煩暑。自是王母三降於燕宫，而昭王狥於攻取，不能遵甘需澄靜之旨。王母亦不復至。甘需曰〔五〕：「王母所設之饌，非人世所有〔六〕。玉酒金醴，後期萬祀，王既嘗之，自當得道矣。但在虛凝純白〔七〕，保其遐齡耳。」甘需亦昇天而去。三十三年，王無疾而殂，形骨柔軟，香氣盈庭，子惠王立矣。{出《仙傳拾遺》〔八〕}
>
> 【校記】
>
> 〔一〕桂 孫本作「柱」。
>
> 〔二〕砂 珠原作「珠砂」，現據孫本、沈本、陳本改。
>
> 〔三〕佩 孫本作「現」。
>
> 〔四〕王登捱日之臺　孫本「王」作「玉」，「捱」作「掘」；沈本「日」作「曰」。
>
> 〔五〕曰 原作「白」，現據孫本改。

〔六〕人世 孫本作「世人之」，沈本作「世人」。

〔七〕凝 原作「疑」，現據沈本改。

〔八〕《拾遺記》卷四、《三洞群仙録》卷九引有此條。

如上，现在的需求是把〔一〕、〔二〕、……，替换成[1]、[2]、……

最常见的方法是查找〔一〕替换成[1]、查找〔二〕替换成[2]、……。如果有几百个条目需要替换则需要替换几百次。而且如果下次又遇到这种情况，仍然需要再重新操作一次，不光浪费时间还很机械。现在我们借助EmEditor的批处理功能，我们只需要预先处理好要查找替换的内容，然后保存一份配置文件。下次再遇到这种情况，就可以直接导入这个配置文件进行批量替换。替换完之后，再把这个导入的配置文件删掉。

具体的操作如下：

启动EmEditor软件，按快捷键「Ctrl＋H」调出「查找替换」窗口，或依次点击「搜索→替换」：

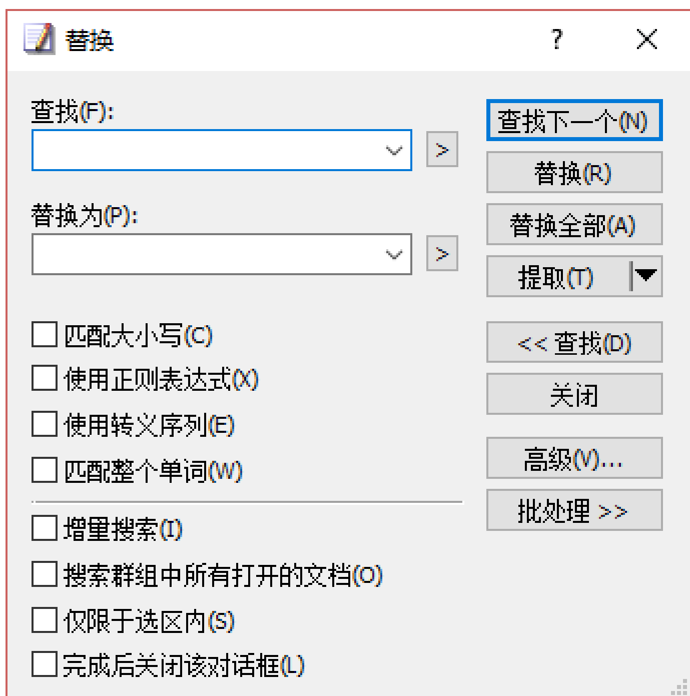


图7-13 「替换」窗口

因为上面的需求是一对一的字面值替换，所以不用勾选「使用正则表达式」，然后点击右下角的「批处理按钮」，窗口变成如下：

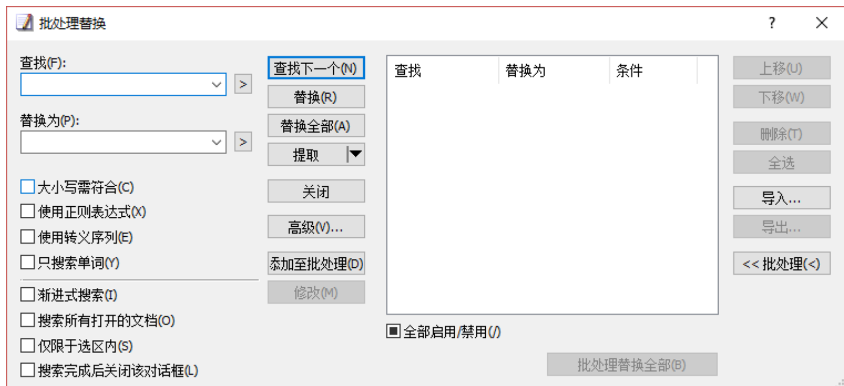


图7-14　「批处理替换」窗口

在左侧的「查找」框中输入要查找的字符，在「替换为」框中输入替换字符，然后点击「添加至批处理」按钮，完成后如下：

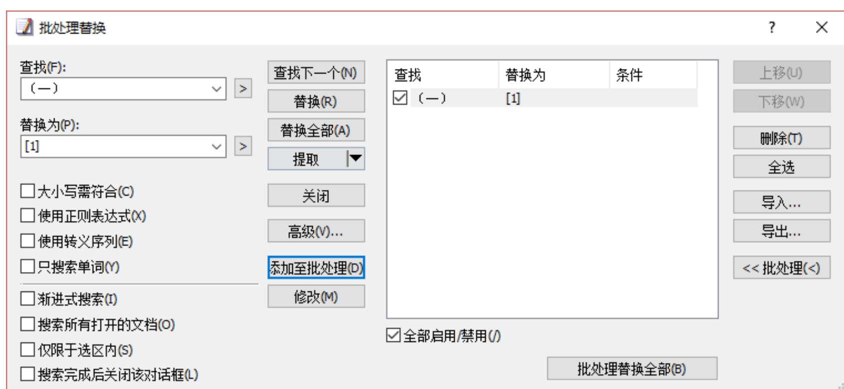


图7-15　「批处理替换」条目

后面需要替换的字符就可以按照上面的步骤操作。还有一种方法也可以添加查找替换内容，点击上图「右侧」一列的「导出」按钮，对导出文件进行重命名。然后选择「记事本」方式打开这个文件：

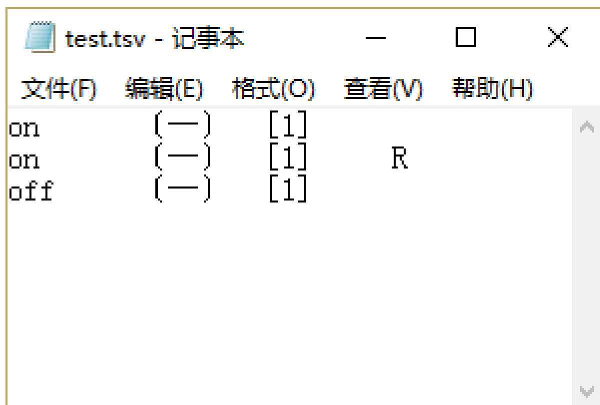


图7-16　导出的替换条目

从上图可以看出来，「on」表示查找处于勾选状态，「〔一〕」表示查找内容，「[1]」表示替换内容，条件为空，即普通字面值查找替换。

需要说明一下，与on对应的是off，即当查找到这个条目出现off时跳过当前条目的匹配；当你勾选「使用正则表达式」后，条件一列会出现R。

其中列与列之间使用制表符隔开，即Tab空格（正则表达式为\t）。根据上面的规则我们就可以在记事本中进行查找替换条目的编辑，完成后如下：

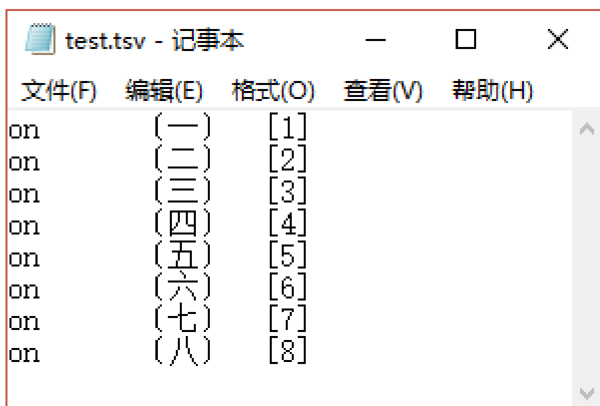


图7-17　编辑替换条目

编辑完之后，按快捷键「Ctrl+S」保存或依次点击「文件→保存」，命名为「number.tsv」。然后回到EmEditor软件中。

之前为了测试我们手动添加了一个查找替换条目，现在你可以点击这条目，然后点击右侧的「删除」按钮，删掉这个条目。接着载入我们刚才编辑的配置文件，点击右侧的「导入」按钮，从弹出对话框中找到这个

文件（number.tsv），然后打开它。完成后如下：

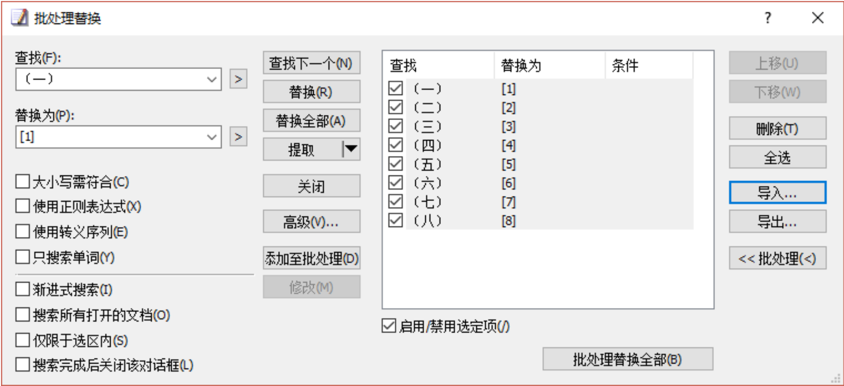


图7-18　导入编辑好的替换条目

可以发现所有的条目都加载好了，现在你可以复制上文中的案例段落，保存为一个文本文档（text.txt），然后用EmEditor打开它，如下：

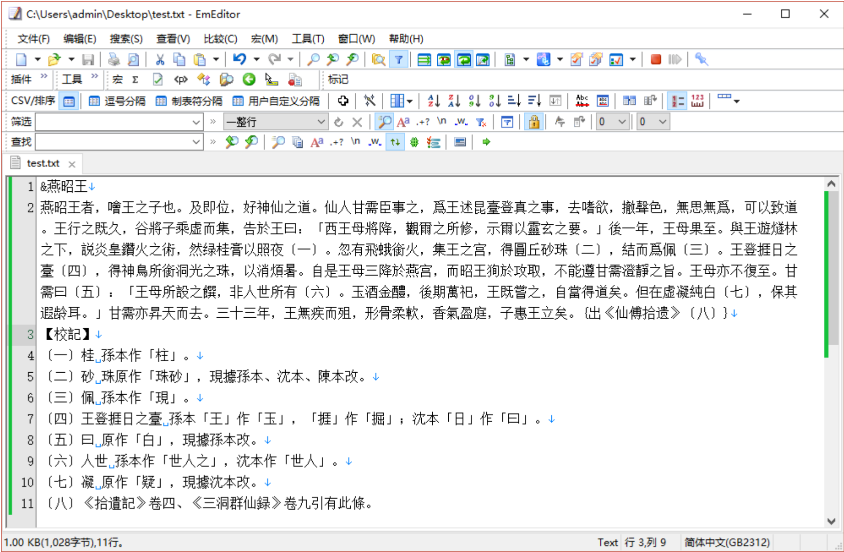


图7-19　案例段落

现在按快捷键「Ctrl＋H」调出替换窗口，点击「批处理替换全部」按钮：

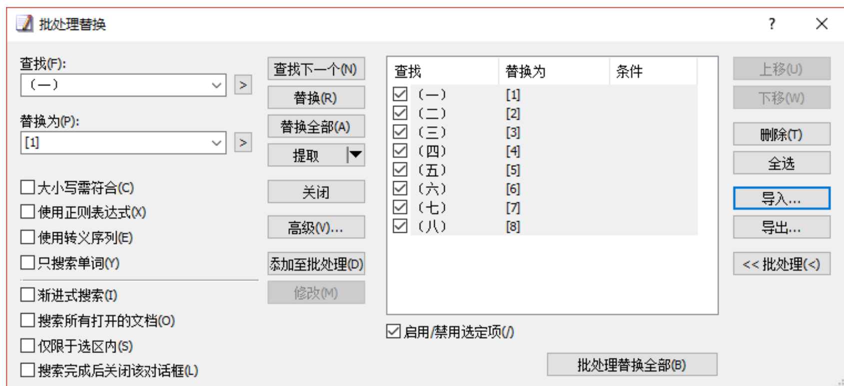


图7-20　运行批处理替换

可以发现这个处理的过程很快，当然这个演示用的文本文档大小也有关系。完成后如下：

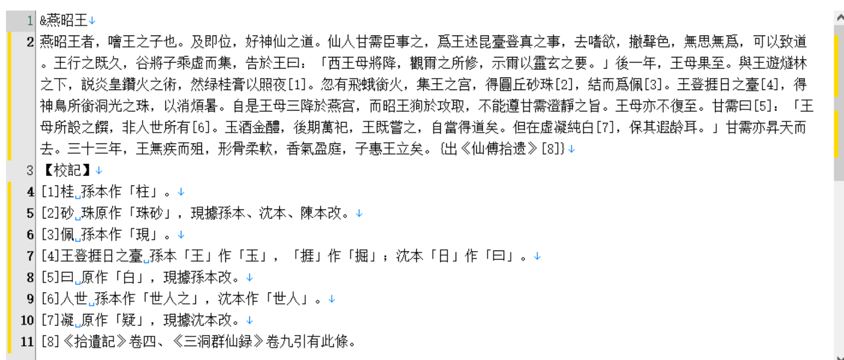

1 &燕昭王
2 燕昭王者，噲王之子也。及即位，好神仙之道。仙人甘需臣事之，爲王述昆臺登真之事，去嗜欲，撤聲色，無思無爲，可以致道。王行之既久，谷將子乘虛而集，告於王曰：「西王母將降，觀爾之所修，示爾以靈玄之要。」後一年，王母果至。與王遊燧林之下，説炎皇鑽火之術，然綠桂膏以照夜[1]。忽有飛蛾銜火，集王之宮，得圓丘砂珠[2]，結而爲佩[3]。王登捱日之臺[4]，得神鳥所銜洞光之珠，以消煩暑。自是王母三降於燕宮，而昭王狥於攻取，不能遵甘需澄靜之旨。王母亦不復至。甘需曰[5]：「王母所設之饌，非人世所有[6]。玉酒金醴，後期萬祀，王既嘗之，自當得道矣。但在虛凝純白[7]，保其遐齡耳。」甘需亦昇天而去。三十三年，王無疾而殂，形骨柔軟，香氣盈庭，子惠王立矣。{出《仙傳拾遺》[8]}
3 【校記】
4 [1]桂 孫本作「柱」。
5 [2]砂 珠原作「珠砂」，現據孫本、沈本、陳本改。
6 [3]佩 孫本作「現」。
7 [4]王登捱日之臺 孫本「王」作「玉」，「捱」作「掘」；沈本「日」作「曰」。
8 [5]曰 原作「白」，現據孫本改。
9 [6]人世 孫本作「世人之」，沈本作「世人」。
10 [7]凝 原作「疑」，現據沈本改。
11 [8]《拾遺記》卷四、《三洞群仙録》卷九引有此條。

图7-21　批处理完成

现在你已经掌握了这个批处理功能使用的步骤了。

实际上这个和Sigil的「搜索模板」功能非常相似，都是预先设置好查找替换条目，然后可以一键替换。下一节讲下如何使用这个功能以来帮助我们校对文本。

7.4 批处理应用（配合多看读校文本）

上一节中讲了EmEditor的批处理功能，本节我就这个功能再配合多看阅读来帮助我们校对文本。

首先，复制上一节中的案例段落，保存为一个文本文档，然后发送到手机中。打开多看阅读APP（如果你还没有安装，请到应用市场或官网下载安装），切换到「书架」界面，点击右上角的菜单图标，选择「导入图书」。在「智能导入书籍」中，点击界面底部的「立即扫描」按钮。待扫描完成后，勾选测试文本，选择「导入书架」，在弹出框中选择「确认」。然后在书架中打开它，如果你觉得TXT格式的排版很丑，可以自定义排版样式，这里不再赘述。

test

&燕昭王

燕昭王者，噲王之子也。及即位，好神仙之道。仙人甘需臣事之，爲王述昆臺登真之事，去嗜欲，撤聲色，無思無爲，可以致道。王行之既久，谷將子乘虛而集，告於王曰：「西王母將降，觀爾之所修，示爾以靈玄之要。」後一年，王母果至。與王遊燧林之下，説炎皇鑽火之術，然绿桂膏以照夜〔一〕，忽有飛蛾銜火，集王之宫，得圓丘砂珠〔二〕，結而爲佩〔三〕。王登捱日之臺〔四〕，得神鳥所銜洞光之珠，以消煩暑。自是王母三降於燕宫，而昭王狥於攻取，不能遵甘需澄靜之旨。王母亦不復至。甘需曰〔五〕：「王母所設之饌，非人世所有〔六〕，玉酒金醴，後期萬祀，王既嘗之，自當得道矣。但在虛凝純白〔七〕，保其遐齡耳。」甘需亦昇天而去。三十三年，王無疾而殂，形骨柔軟，香氣盈庭，子惠王立矣。{出《仙傳拾遺》〔八〕}

【校記】

〔一〕桂 孫本作「柱」。

〔二〕砂 珠原作「珠砂」，現據孫本、沈本、陳本改。

〔三〕佩 孫本作「現」。

〔四〕王登捱日之臺 孫本「王」作「玉」，「捱」作「掘」；沈本「日」作「曰」。

〔五〕曰 原作「白」，現據孫本改。

〔六〕人世 孫本作「世人之」，沈本作「世人」。

〔七〕凝 原作「疑」，現據沈本改。

〔八〕《拾遺記》卷四、《三洞群仙録》卷九引有此條。

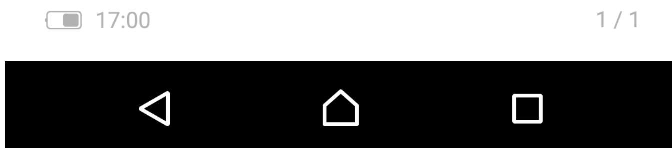


图7-22　多看阅读中记录笔记

长按屏幕，选择查找内容，在弹出框中选择「划线」（第1张图），然后再选择「笔记」（第2张图）。界面会切换到编辑界面（第3张图），此时你可以输入替换内容，内容编辑好之后，点击右上角的「完成」按钮。剩余项如法操作.完成后如第4张图。

回到「书架」界面，点击右下角的「我的」图标，下翻找到「笔记」行，找到测试的书籍。点击进入详情页，可以看到我们刚才编辑的笔记在这里会完整展示，有笔记内容和标题条数等。

回到「笔记」页面顶部，点击「共8条笔记」的最右侧的分享图标，在弹出框中选择「其他」，然后选择「保存到QQ收藏」，在弹出窗口中点击「确认」按钮。

电脑上登录QQ，在QQ面板的左下角点击菜单图标，选择「我的收藏」，在弹出框中选择左上角的「同步」按钮，同步你的移动端的收藏。

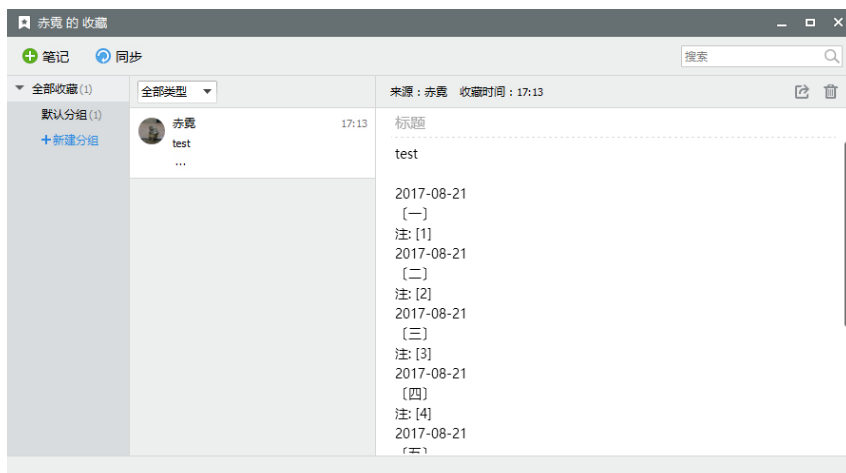


图7-23　导出笔记内容

按快捷键「Ctrl＋A」全选笔记内容，「Ctrl＋C」复制选择的文本，在EmEditor中粘贴（Ctrl＋V）复制的内容。

删掉第一行中的书名信息，然后使用下面的正则表达式把这个笔记内容格式改成批处理模板（number.tsv）一样的格式，见上一节。可复制下面的内容到文本文档中，然后重命名为「test-note」，扩展名修改为「.tsv」：

```
on ^\d{4}-\d{2}-\d{2}\n(.*?) on\t\1 R
on \n注: (.*?) \t\1 R
```

然后把它导入到EmEditor中，批处理笔记内容。处理好的笔记内容，保存为tsv格式（此为处理测试文本的批处理tsv文件）。

然后删掉处理笔记内容的批处理条目，导入处理测试文本的批处理tsv文件。执行「批处理替换全部」，然后保存测试文本。

> 笔记内容与划线内容一定是一对一的修改，且尽量多的去选择划线内容，一般3-5字符为宜，这可以避免替换错！

7.5 宏脚本

宏脚本与批处理模板类似，不过比批处理模板更强大。下面演示一个比较简单的批处理模板（由好友 @拐子 提供）。

安装脚本

复制下面的代码，粘贴到一个文本文档中，重命名为「批处理模板.jsee」：

```
var dict={"查找内容":"替换内容",
"查找内容":"提换内容"};
for(key in dict){
    var
status=document.selection.Replace(key,dict[key],eeReplaceAll);
}
```

然后使用正则表达式编辑「查找内容」和「替换内容」，完成后如下：

```
var dict={"〔一〕":"[1]",
"〔二〕":"[2]",
"〔三〕":"[3]",
"〔四〕":"[4]",
"〔五〕":"[5]",
"〔六〕":"[6]",
"〔七〕":"[7]",
"〔八〕":"[8]"};
for(key in dict){

  var status=document.selection.Replace(key,dict[key],eeReplaceAll
}
```

另存为一个文件，选择一个文件夹保存该模板，命名为「模板.jsee」。

再复制7.3的案例段落，保存为一个TXT文本（test.txt）。然后开始安装脚本文件，依次点击「宏→选择」：

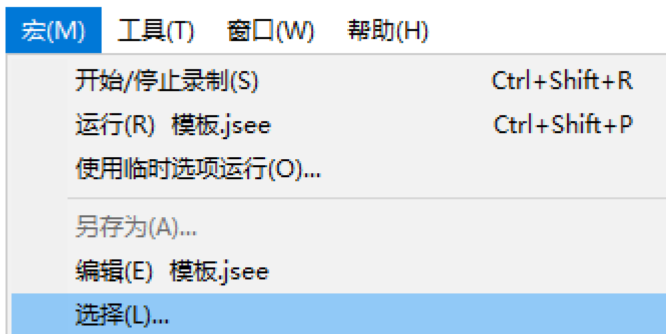


图7-24　安装脚本

在弹出对话框中选择上一步保存的模板，然后打开它。接着打开案例TXT文本（test.txt），依次点击「宏→模板.jsee」。软件会自动执行查找替换条目，并依次替换。

如果需要编辑脚本，可以点击「宏→编辑...」，进入脚本标签页，然后和编辑文本一样编辑它。

删除脚本

删除一个脚本，依次点击「宏→自定义」：

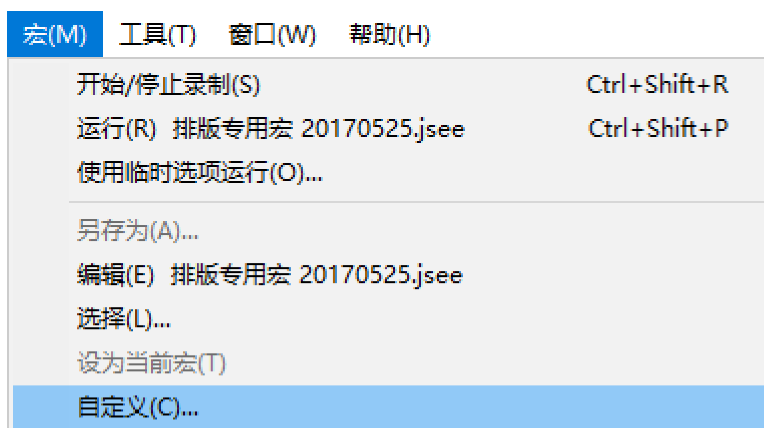


图7-25　脚本自定义

在弹出窗口中，先选择要删除的脚本，然后点击右侧的「删除」按钮。

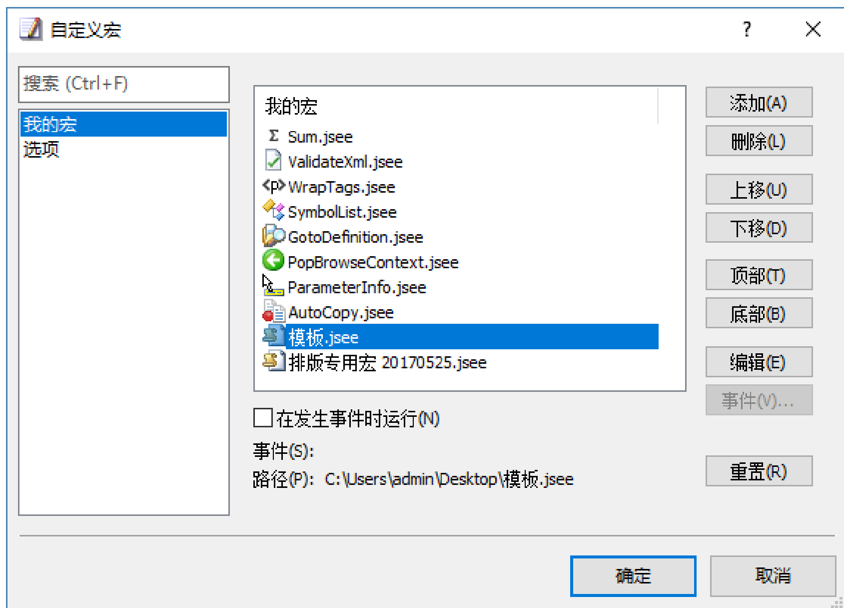


图7-26　编辑处理脚本文件

删除完成后，点击右下角的「确定」退出该窗口。如果你有需要，你也可以对脚本进行上下排序。

使用脚本

下载精校组排版脚本.jsee，然后导入该脚本，依次点击「宏→选择」，找到这个脚本，打开它。使用这个脚本时，依次点击「宏→精校组排版脚本.jsee」，会弹出如下：

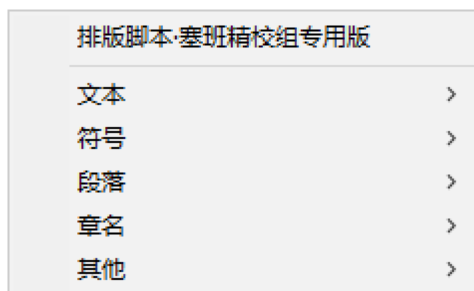


图7-27　运行处理脚本

根据你的需求，选择相应的菜单，击右侧的三角，展开。这里以段落为例：

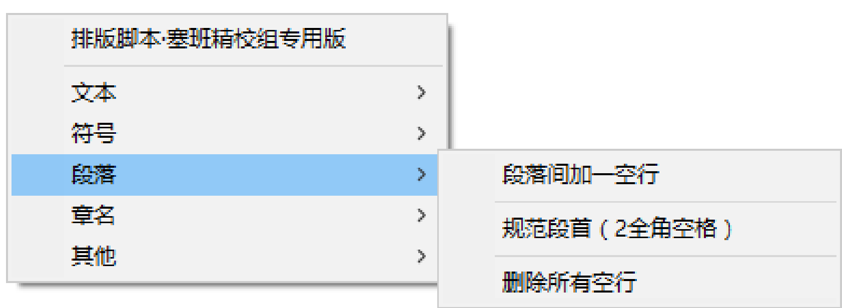


图7-28　执行脚本处理

点击「段落间加一空行」，会为处理的文本文件每段添加一个换行符（\n），显示效果即空行。点击「规范段首（2全角空格）」，会在每段的首行开始的位置添加2个全角空格，这与CSS中的`text-indent: 2em;`效果一致，但实现方式不同。「删除所有空格」会删除段与段之间的换行符。以上三个功能实际就是正则表达式的应用：

段落间加一空行

```
查找：$
替换：\n
```

规范段首（2全角空格）

```
查找：^[　 ]+$、[　　 　　　]+$
替换为空
查找：^([^　 ])
替换：　　\1
查找：^[　]+
替换：
查找：^[ ]{3,8}
替换：
```

删除所有空格

```
查找：^[　 ]*\n
替换为空
```

关于这个脚本的更多使用功能，可自己摸索。

这里还有一个排版专用宏 20170525.jsee，也可以研究下。

7.6　大纲栏

EmEditor自带大纲插件，经过设置后可以用来显示目录。对于TXT格式小说或者比较大的文件，可以方便地定位。

依次点击「查看→大纲栏」、「查看→大纲导航」，让这两个选项处于选择状态：

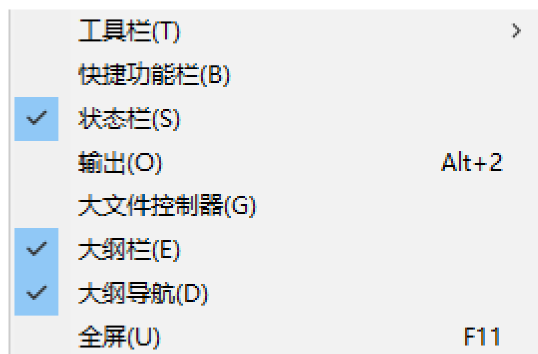


图7-29　勾选「大纲栏」和「大纲导航」

默认大纲栏是显示在右侧的，一般我会设置大纲面板显示在左侧，具体设置方法如下：

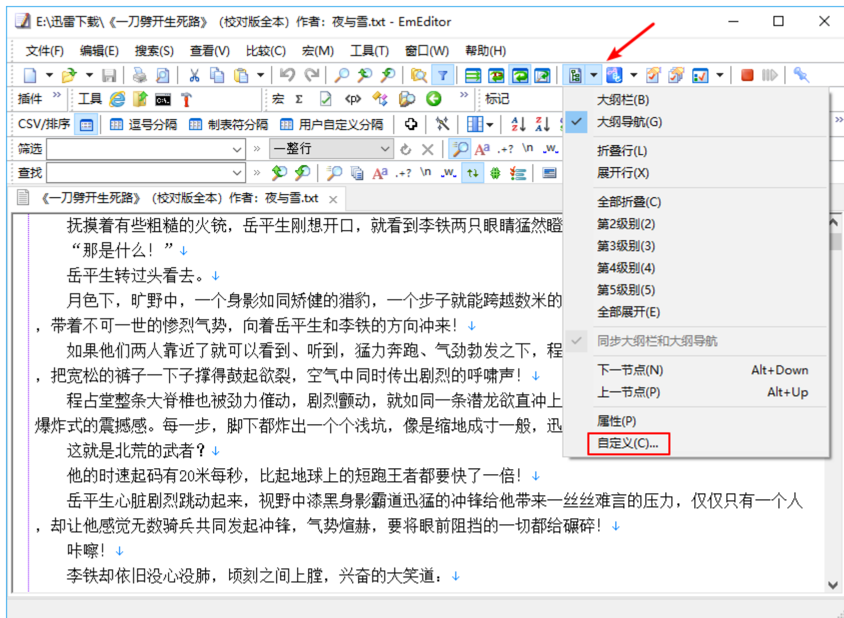


图7-30　设置大纲面板显示位置

点击红色箭头所指的三角形，会弹出一个菜单列表，点击红框中得「自定义」选项。会弹出下面的对话框：

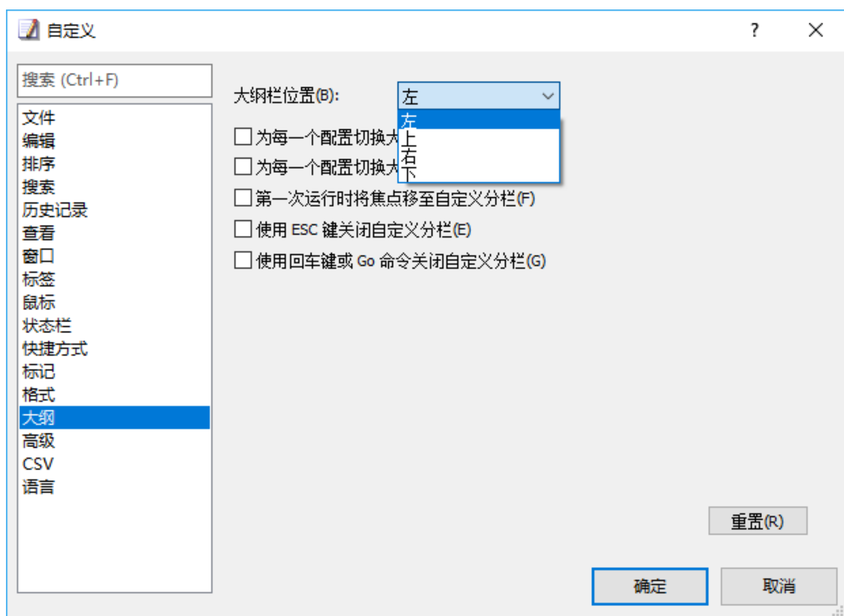


图7-31　大纲设置面板

现在你可以设置大纲栏显示的位置了，点击「大纲栏位置」右侧的向下箭头，选择要显示的位置。完成后，点击「确定」按钮，保存设置。

然后在左侧的大纲栏空白处点击鼠标右键，会弹出下面的菜单选项：

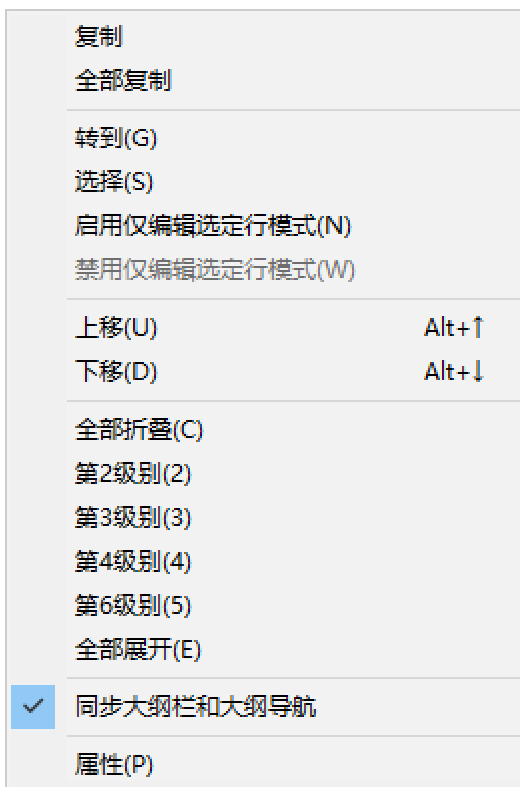


图7-32　大纲栏代码设置

点击最底部的「属性」，弹出下面的设置面板：

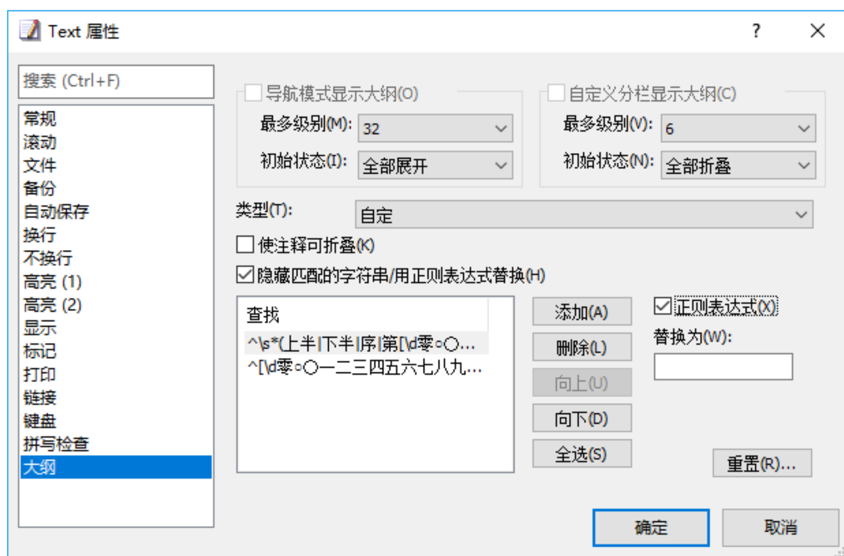


图7-33　标题的正则表达式

首先「类型」选项选择「自定」，取消勾选「使注释可折叠」，勾选「隐藏匹配的字符串/用正则表达式替换」。点击「添加」按钮，分别复制下面的代码，添加到「查找」项中：

```
第一项：^\s*(上半|下半|序|第[\d零○〇一二三四五六七八九十百千]{0,10})(卷|章|回|话|集|部).{0,40}$
第二项：^[\d零○〇一二三四五六七八九十百千]{1,10}(卷|章|回|话)?[\s|、].{0,30}$
```

同时注意不要忘记勾选「正则表达式」。然后点击「确定」按钮保存设置即可。

现在可以看到在左侧大纲栏，显示了小说中的标题了。点击标题，就可以跳转到对应的章节中了。

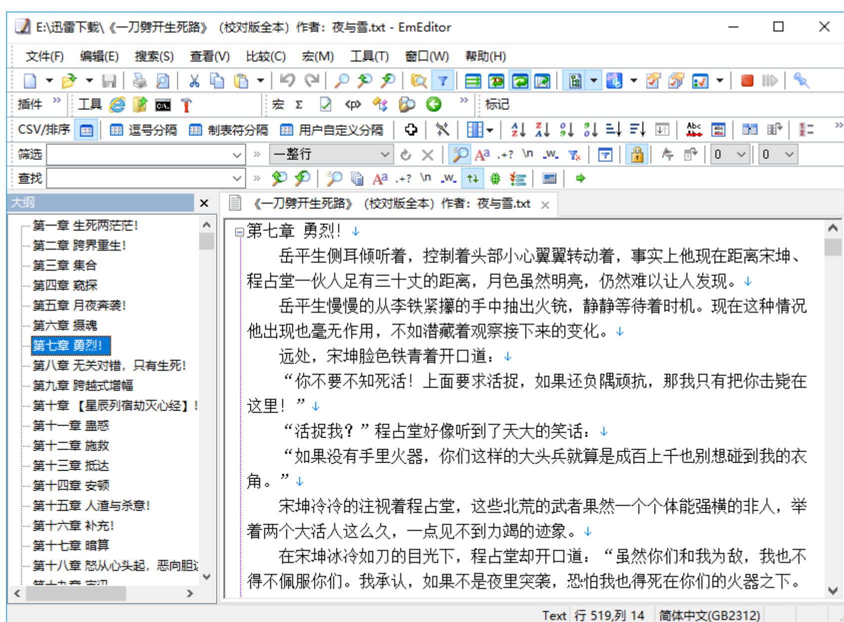


图7-34　大纲栏效果

本文参考百度贴吧的这个帖子《EmEditor的大纲功能很便利》，感谢原文作者 @刹那星劫 提供的正则表达式！

第8章

制作电子书的辅助工具

制作ePub电子书是个比较复杂的工作，比如按照资源类型来说，有PDF、Docx、TXT、Djvu等格式，每种格式的打开软件又不相同。按照功能来说，光一个子集字体的工具就有3种，然后查看字体文件还需要用到Fontlab。ePub转格式又有多种要求，转mobi、转PDF等。

以上情况的出现，都要求我们需要对这些软件尽量熟悉，以便快速地处理遇到的问题。

8.1 Adobe Acrobat——编辑PDF

Adobe Acrobat是由Adobe Systems所开发的电子文字处理软件集，可用于阅读、编辑、管理和共享PDF格式文档。

在6.3.1一节中，就使用了Acrobat的部分功能。下文从PDF加书签和OCR两方面来讲一下Acrobat的使用，本次演示的版本为Adobe Acrobat Pro DC 2017。软件可以从大眼仔旭网站获得，这里不提供下载链接。

8.1.1 添加书签

因为PDF的便利性，现在网络上有海量的PDF资源，但是大部分PDF没有书签，导致阅读时不甚方便。且加了书签的PDF文件，在我们制作ePub时，对于查阅对比非常有帮助。

本次演示案例以2006年上海古籍出版社的《中国小说史略》为例，你可以下载案例PDF，跟着教程一步步学习。在制作前还需要下载两个软件：书签获取软件和PdgCntEditor（老马的原创空间＞原创软件＞PdgCnEdtor_chn.zip）。

打开全国图书馆参考咨询联盟网站，见6.3.1节内容。到预览界面：



图8-1　获取图书唯一ID

复制这串数字，打开「书签获取软件V0505.exe」，界面如下：

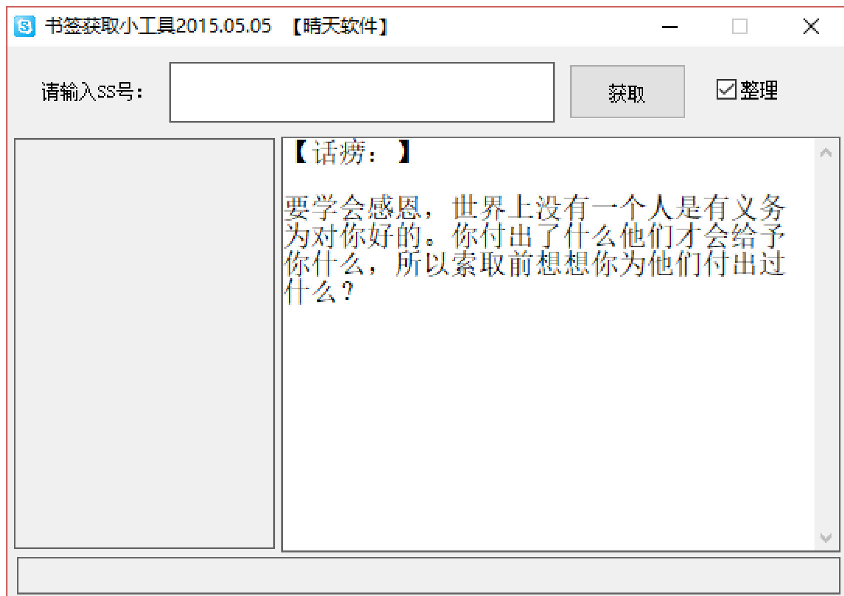


图8-2　下载图书的目录信息

粘贴刚才复制的数字编号到「请输入SS号」输入框中，然后软件将会自动获取该书的目录信息，信息获取成功后如下：

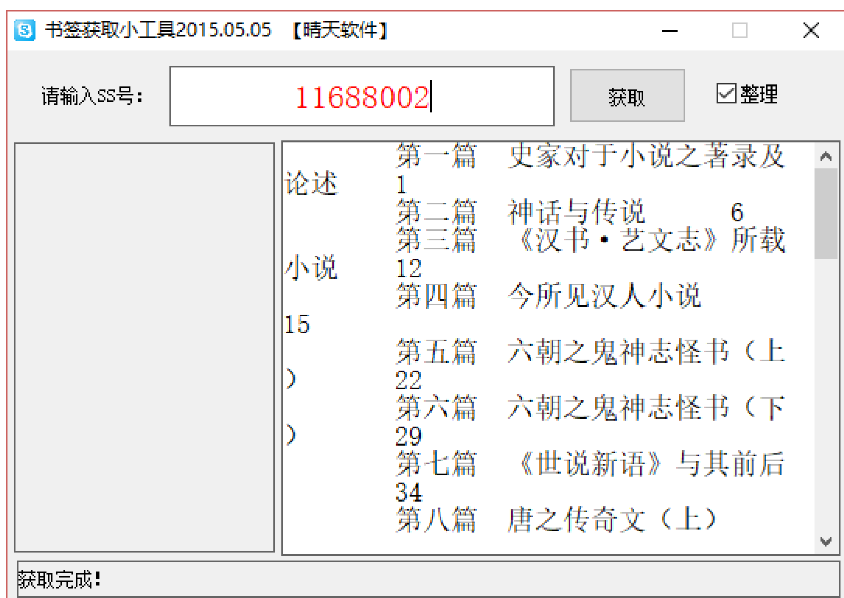


图8-3　复制图书目录信息

这时你可以先打开PdgCnEdtor软件，然后使用该软件打开下载好的案例PDF——《中国小说史略》，点击左上角「打开」图标，在弹出对话框中选择PDF，然后打开它：

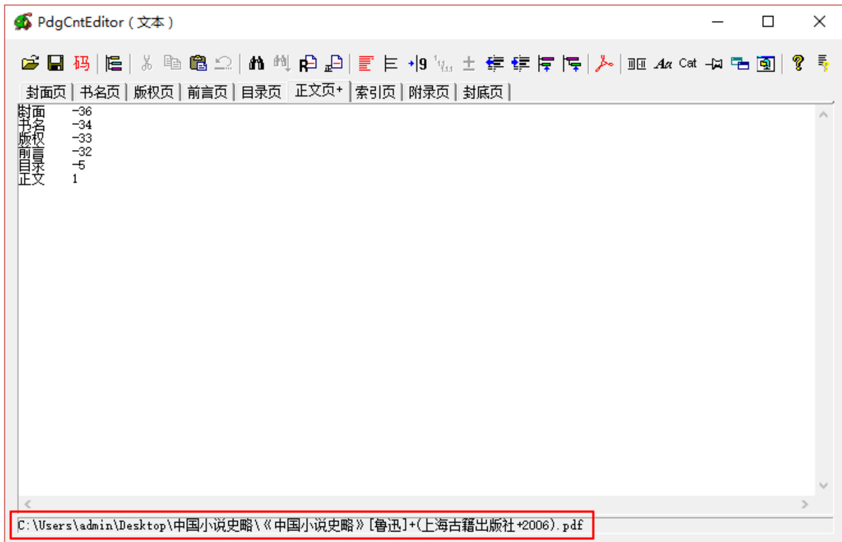


图8-4　打开需要编辑书签的PDF文件

复制上一步软件获取的目录信息到PdgCnEdtor软件的编辑窗口中：

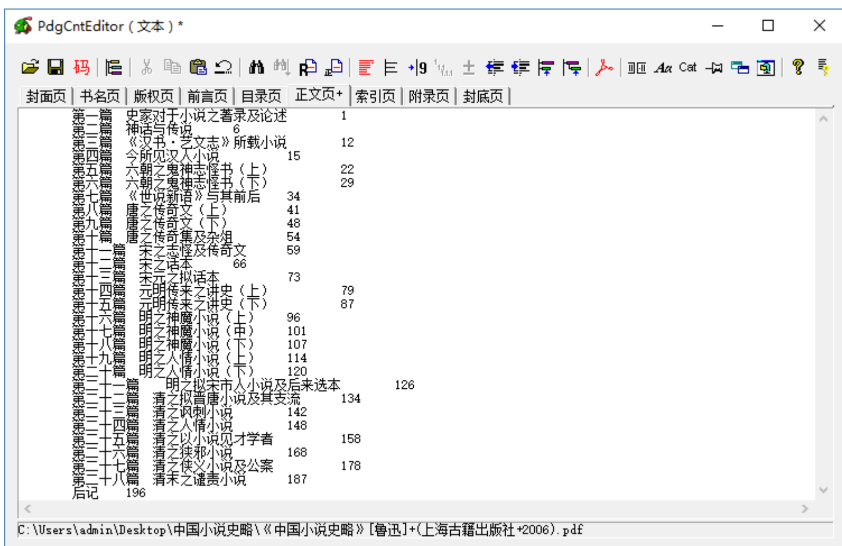


图8-5　粘贴图书目录信息

然后点击左上角第二个「保存」图标。

> 用PdgCnEdtor保存时，被编辑的PDF不要在任何软件中打开！

上述操作完成后，关闭PdgCnEdtor，再用Acrobat打开电子书，如下所示：

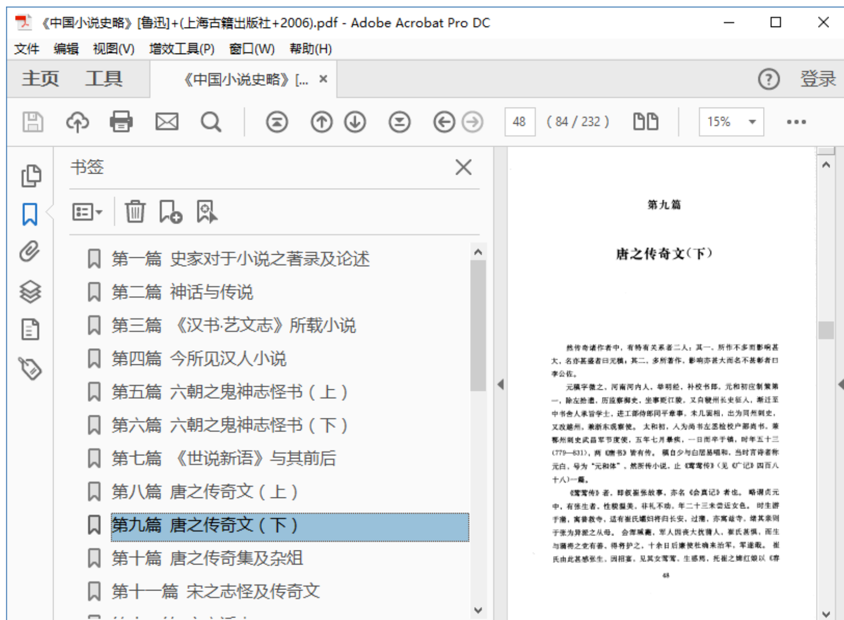


图8-6 测试书签的链接目标

点击左侧的书签列表即可以跳转到该书签所链接的章节，你可以把每个书签都点击一遍，确认所有的书签都链接了章节。到这一步，我们还需要再做一点优化工作，按快捷键「Ctrl＋D」，弹出下面的对话框。

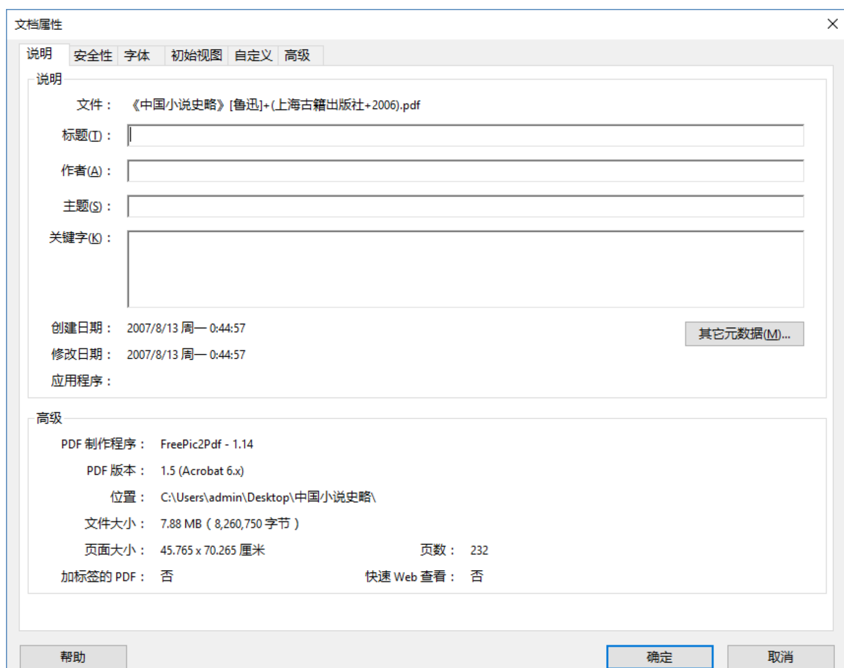


图8-7 文档属性说明

在「说明」和「初始视图」页面中填写/设置如下：

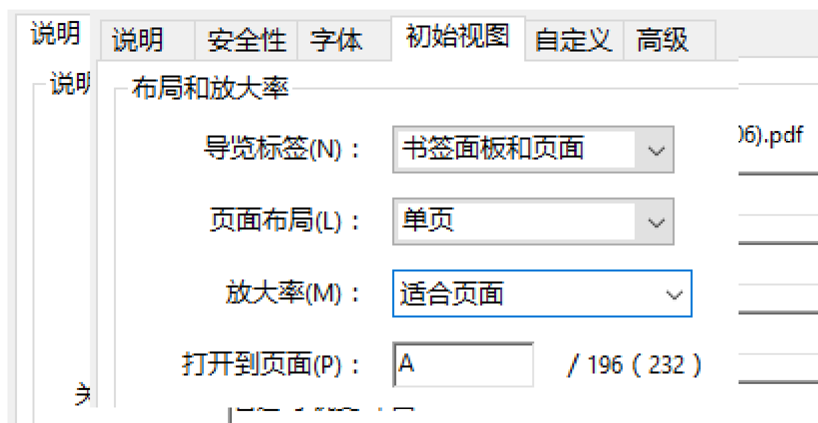


图8-9　设置初始视图

图8-8　填写文档属性说明

然后点击右下角的「确定」按钮，关闭Acrobat软件时保存PDF。重启Acrobat，再打开PDF文件。初始视图将会按照你设置的显示。

上面演示的是只有一个层级的标题书签，但实际上有的书目录页有多个层级，比如一级、二级、三级标题。这种情况下，我们需要使用PdgCnEdtor编辑目录层级。

仍然以《中国小说史略》为例，我们先在Acrobat中把PDF的辅助页书签添加进去。

打开PDF文件，第一页显示的是封面页，按快捷键「Ctrl＋B」为当前页添加书签。如果你在PDF打开后没有点击过书签，执行上面的添加快捷键，会把未命名书签添加到书签列表的底部。为了避免这种问题的发生，你可以点击第一个书签，把页面再向上滚动到第一页——封面页。

然后再按快捷键「Ctrl＋B」添加封面页书签，如下：

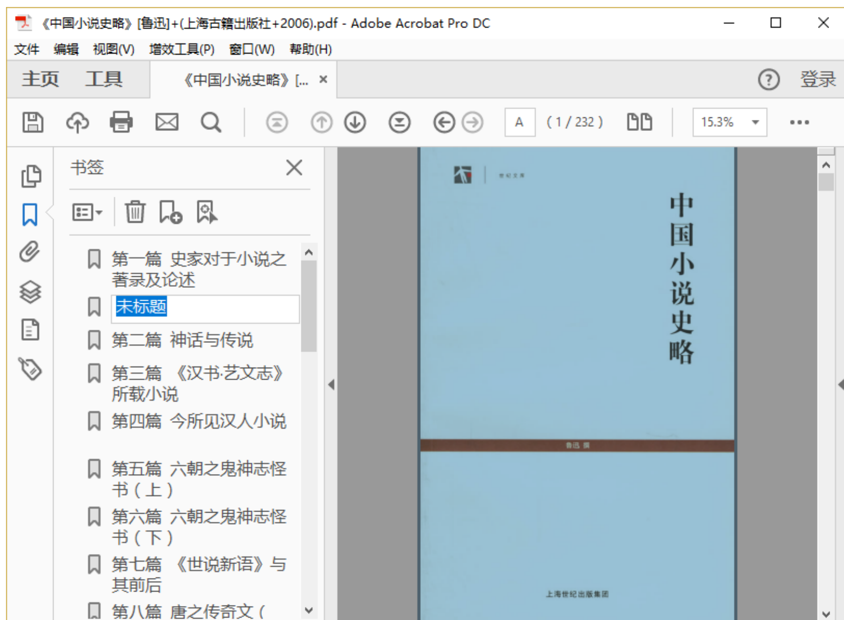


图8-10　手动添加书签

修改「未标题」为「封面」，然后按Enter键确认。然后选择「封面」书签，点击鼠标左键，向上移动如第1张图，然后松手。：

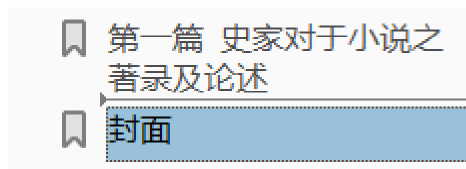


图8-12　二级标题

这一步操作往往会出现两种情况，稍不留神就会出错，所以应仔细分辨两种不同的提示符号。第1张图表示的是与「第一篇　史家...」为同一级别，第2张图表示的是「第一篇　史家...」的子级标题。下面的扉页、版权页、前言以及其下的二级标题、目录依次按照上述步骤添加，最后完成如下：

封面

封底

扉页

版权页

出版说明

前言

题记

序言

目录

图8-13　修改好的书签

我们再使用PdgCnEdtor打开这个PDF文件，可以发现在二级标题前是有一个Tab空格的：

PdgCntEditor（文本）

封面页 | 书名页 | 版权页 | 前言页 | 目录页 | 正文页+ | 索引页 | 附录页 | 封底页

```
封面	-36
封底	-35
扉页	-34
版权页	-33
出版说明	-30
前言	-27
	一	-27
	二	-20
题记	-12
序言	-11
目录	-5
第一篇	史家对于小说之著录及论述	1
第二篇	神话与传说	6
第三篇	《汉书·艺文志》所载小说	12
第四篇	今所见汉人小说	15
第五篇	六朝之鬼神志怪书（上）	22
第六篇	六朝之鬼神志怪书（下）	29
第七篇	《世说新语》与其前后	34
第八篇	唐之传奇文（上）	41
第九篇	唐之传奇文（下）	48
第十篇	唐之传奇集及杂俎	54
第十一篇	宋之志怪及传奇文	59
第十二篇	宋之话本	66
第十三篇	宋元之拟话本	73
第十四篇	元明传来之讲史（上）	79
第十五篇	元明传来之讲史（下）	87
第十六篇	明之神魔小说（上）	96
第十七篇	明之神魔小说（中）	101
第十八篇	明之神魔小说（下）	107
```

C:\Users\admin\Desktop\中国小说史略\《中国小说史略》[鲁迅]+(上海古籍出版社+2006).pdf

图8-14　PdgCnEdtor编辑视图

从上面的目录中可以看出，一级标题前不存在任何形式的空格，二级标题前增加了一个Tab空格，表示的即是二级标题。同理，如果有标题需作为三级标题，那么只需要在要做三级标题的前面增加两个Tab空格即可，至于四级、五级标题，依此。

8.1.2 文字识别（OCR）

光学字符识别（英语：Optical Character Recognition, OCR）是指对文本资料的图像文件进行分析识别处理，获取文字及版面信息的过程。

识别过程

图像输入、图像预处理：

图像输入：将不同格式的图片输入到OCR软件中，常见图片格式有jpg、png等，因为扫描出来的pdf文件实际上也是图片文件，所以也可以直接导入。

预处理：主要包括二值化，噪声去除，倾斜校正等。

二值化：

对摄像头拍摄的图片，大多数是彩色图像，彩色图像所含信息量巨大，对于图片的内容，我们可以简单的分为前景与背景，为了让计算机更快的、更好地识别文字，我们需要先对彩色图进行处理，使图片只剩下前景信息与背景信息，可以简单的定义前景信息为黑色，背景信息为白色，这就是二值化图。通俗点讲，就是把彩色照片变成黑白照片。

噪声去除：

对于不同的文档，我们对噪声的定义可以不同，根据噪声的特征进行去噪，就叫做噪声去除。噪声可以理解为「妨碍人们感觉器官对所接收的信源信息理解的因素」，而图像中各种妨碍人们对其信息接受的因素即可称为图像噪声。简单理解，就是让图片变得更清晰，减少文字识别的干扰。

倾斜校正：

由于一般用户在拍照或扫描文档时，都比较随意，因此获得的图片就不可避免产生倾斜，这就需要文字识别软件进行校正。

版面分析：

将文档图片分段落，分行的过程就叫做版面分析，由于实际文档的多样性、复杂性，因此，目前还没有一个固定的、最优甚至于分字符的分析切割模型。

字符切割：

由于拍照条件的限制，经常造成切割后的图片字符粘连、断笔，因此极大限制了识别系统的性能。

字符识别：

这一研究已经是很早的事情了，比较早有模板匹配，后来以特征提取为主，但是由于文字的位移，笔画的粗细、断笔、粘连、旋转等因素的影响，极大造成了特征提取的难度。

版面还原：

人们希望识别后的文字，仍然像原文档图片那样排列着，段落不变，位置不变，顺序不变地输出到Word文档、PDF文档等，这一过程就叫做版面还原。

后处理、校对：

根据特定的语言上下文的关系，对识别结果进行校正，就是后处理。

实际案例

以《中国小说史略》的正文第1页为例，我们使用Acrobat的文字识别功能。依次点击「视图→工具→增强扫描→打开」，完成后在顶部工具栏中显示如下：

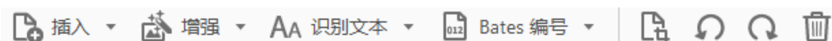


图8-15　增强扫描

点击「识别文本」，选择「在本文件中」选项：

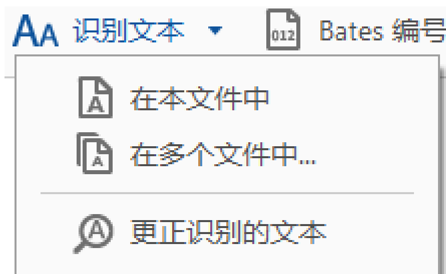


图5-16　识别文本范围

在弹出工具栏中，点击「设置」选项，在弹出窗口中设置如下：

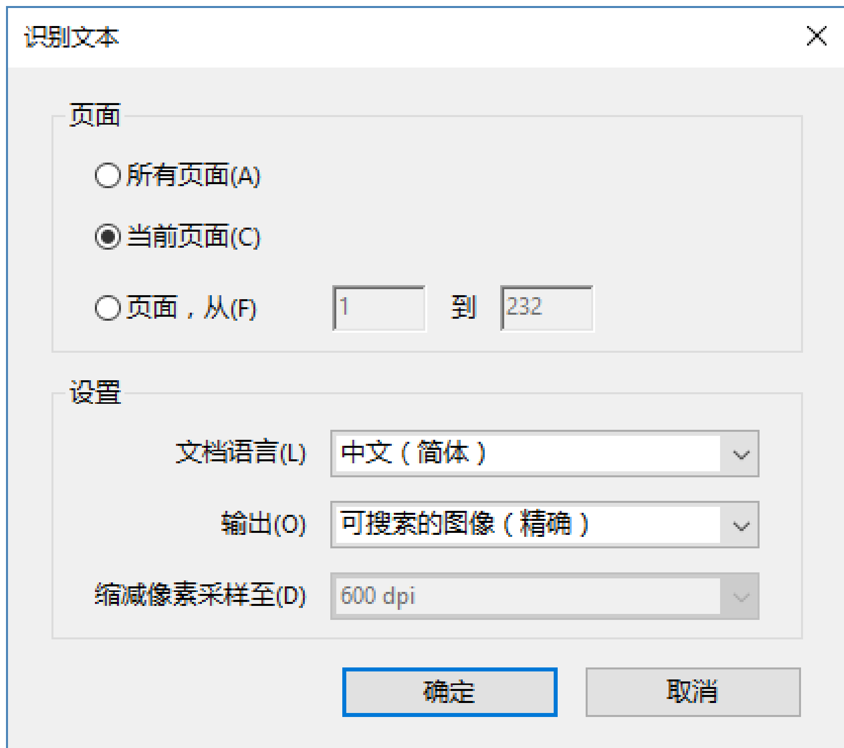


图8-17　设置文本识别范围

然后点击底部的「确定」按钮，这个操作会退出对话框。然后点击工具栏中的「识别文本」：



图8-18　识别文本工具栏

完成后，在当前页面会生成一个文字图层。页面看起来上与未OCR前一模一样。不同之处在于你可以把光标插入到文字当中，选取文字，进行复制。比如：



图8-19　识别后的PDF

在文本编辑器中粘贴所选文本，剩下的就是对照PDF，进行文本的校对了。

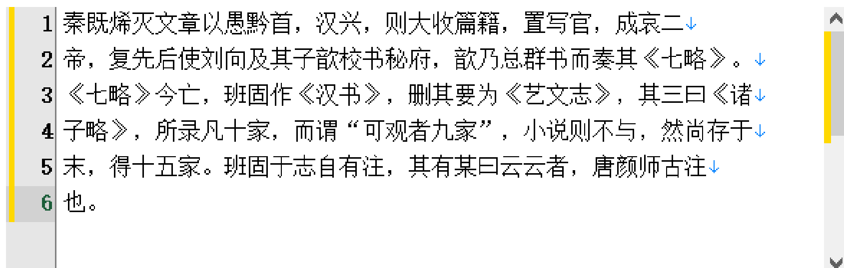

秦既燔灭文章以愚黔首，汉兴，则大收篇籍，置写官，成哀二
帝，复先后使刘向及其子歆校书秘府，歆乃总群书而奏其《七略》。
《七略》今亡，班固作《汉书》，删其要为《艺文志》，其三曰《诸
子略》，所录凡十家，而谓“可观者九家”，小说则不与，然尚存于
末，得十五家。班固于志自有注，其有某曰云云者，唐颜师古注
也。

图8-20　识别后的文本

文字识别的质量取决于你的PDF质量和对PDF的二值化处理。关于这种图像式PDF如何制作成ePub，见12.4节。

8.2 pagina-EPUB-Checker——验证ePub

虽然ePub标准并不难，但其XML文件必须符合特定的模式才算一个标准ePub文件。EpubCheck是一个验证ePub文件的工具。它可以检测出在ePub中的多种错误，OPF、xhtml、CSS等。该验证工具官方下载地址——EpubCheck 4.0.2。

另一种ePub文件验证工具——pagina-EPUB-Checker。此工具基于EpubCheck 4.0.2开发的，相比于官方验证工具而言，它更方便使用。

点开上面的下载链接，下翻找到如下链接，根据你的系统，选择相应的软件包。这里以Windows版本为例：

WINDOWS-DOWNLOAD

Version 1.7.1 (14.12.2016)

Für 32- und 64-bit Java-Versionen: */ For 32- and 64-bit Java versions:*

› pagina-EPUB-Checker_Win-1.7.1.zip (10.7 mb)

MAC OS X-DOWNLOAD

Version 1.7.1 (14.12.2016)

› pagina-EPUB-Checker_Mac-1.7.1.zip (11.3 mb)

LINUX-DOWNLOAD

Version 1.7.1 (14.12.2016)

› pagina-EPUB-Checker_Linux-1.7.1.tar.gz (10.6 mb)

图8-21　pagina-EPUB-Checker的不同版本

点击「pagina-EPUB-Checker_Win-1.7.1.zip」链接，下载文件，解压下载的软件压缩包，会有一个「EPUB-Checker.exe」文件，双击或点击鼠标右键选择「打开」。可能会弹出窗口如下：

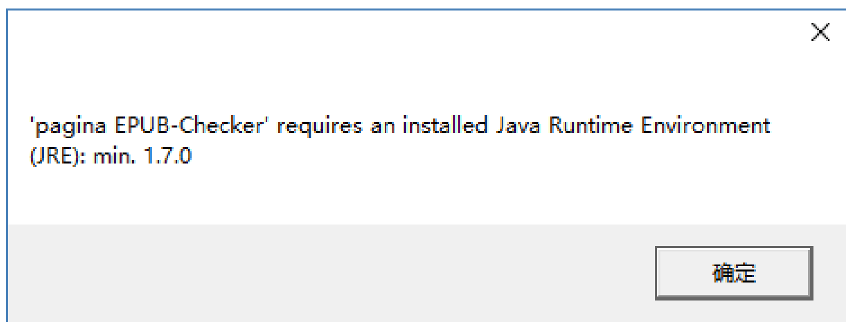


图8-22　提示安装JRE

软件提示需要安装Java环境才可以运行，点击「确定」，会在浏览器中打开一个Java环境下载界面。点击「免费Java下载」按钮：

下载适用于 Windows 的 Java

推荐 Version 8 Update 144 (文件大小： 721.06 KB)

发行日期 2017 年 7 月 26 日

同意并开始免费下载

下载 Java 即表明您确认阅读过并接受 最终用户许可协议的条款

» 什么是 Java？ » 我有 Java 吗？ » 是否需要帮助？

图8-23　下载Java

跳转到新页面，点击「统一并开始免费下载」按钮。

双击或点击鼠标左键选择「打开」下载的文件——JavaSetup8u144.exe。在权限控制对话框中点击「是」按钮，然后弹出如下：



图8-24　安装Java

点击「安装」按钮，等待程序下载文件并安装。完成后会提示你已经安装完成了，然后退出对话框就行。

再次打开「EPUB-Checker.exe」，发现现在可以使用了。下面是软件打开后的界面：

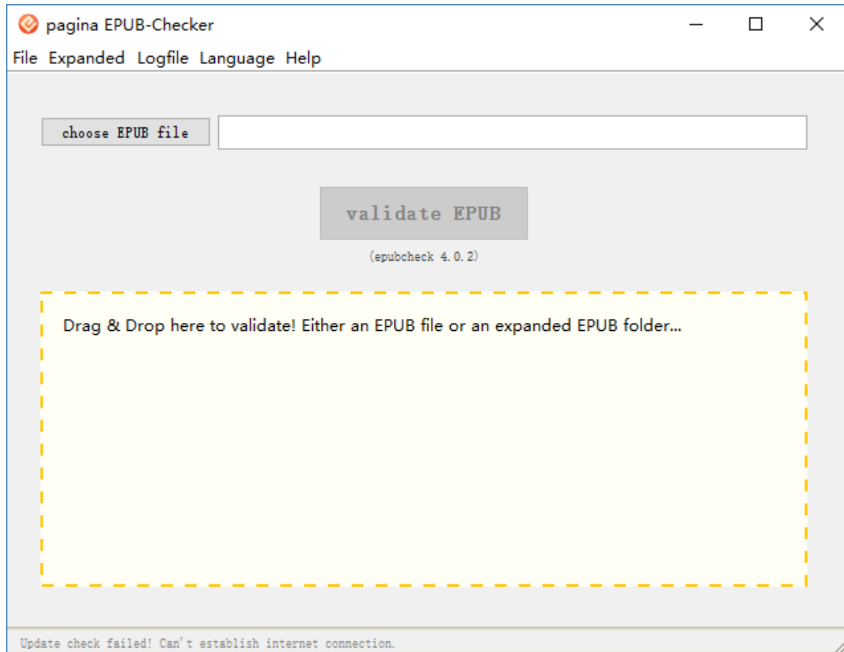


图8-25 pagina-EPUB-Checker操作界面

虽然这个软件有「界面语言」设置选项，但是没有中文选项，所以我们只能在英文界面下使用，不过只是验证ePub的话，对我们没有太大影响。下面以本书为例来体验下这个软件的使用。

点击「choose EPUB file」按钮或依次点击「File→Open」，在弹出对话框中选择要验证的ePub文件。文件打开后会在左下角状态栏中显示如下：

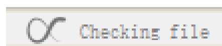


图8-26 正在验证...

这表示软件正在检查ePub文件，检查完成后会以列表的形式显示检查报告。然后你可以根据报告，进行修改。

报告将会以表格的形式体现，当然你也可以设置成文本模式。设置方法是：依次点击「Logfile→Text view」，然后等待软件切换到设置的模式，这可能会等待十几秒，不要着急。这里的演示以默认的表格视图为例。ePub验证后显示如下：

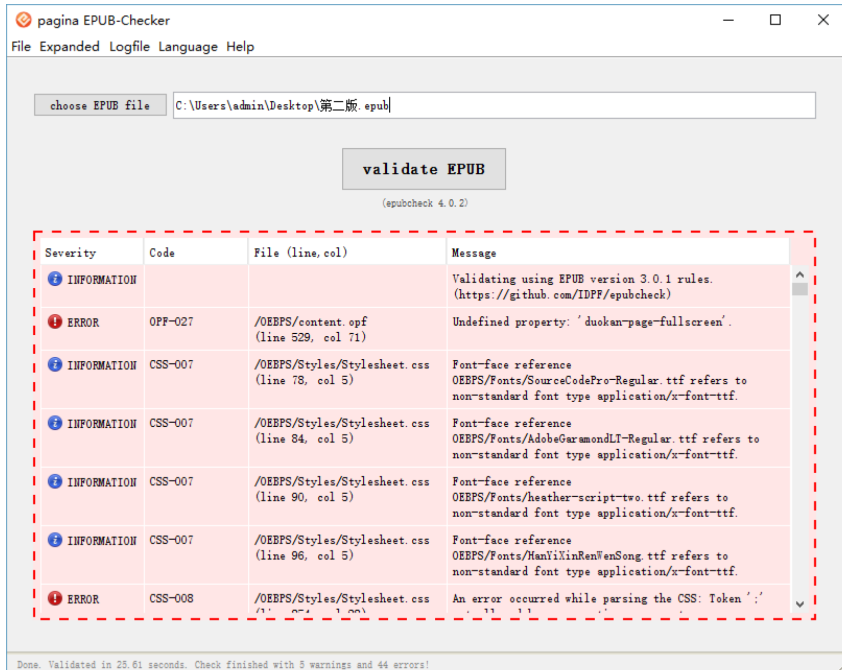


图8-27　验证信息列表

底部状态栏显示本次验证共用时25.61秒，发现5个警告级别和44个报错级别的错误。我们先看第一条报错信息：

Severity	Code	File (line, col)	Message
INFORMATION			Validating using EPUB version 3.0.1 rules. (https://github.com/IDPF/epubcheck)
ERROR	OPF-027	/OEBPS/content.opf (line 536, col 71)	Undefined property: 'duokan-page-fullscreen'.

图8-28　详细验证信息

首先是第1列的「Severity（严重性）」，这里显示为「ERROR（错误）」，提示你这是一个比较重要的错误。然后是第2列的「Code（错误代码）」显示为「OPF-27」，表示的是关于「content.opf」文件的错误，编号为027。第3列「File　(Line,col)」，意思是「文件路径（行，列）」，告诉你在什么文件，文件中的具体坐标。第4列「Message（信息）」显示报错的具体描述信息。

现在根据第3列的提示信息，在Sigil中的「图书浏览器」底部双击打开「content.opf」文件，根据定位（536行）找到报错位置（这个行号可能会变动，因为后面会增加图片，然后资源清单列表同时会增加，导致行号也会变动，请以实际测试的文件为准）：

```
<spine toc="ncx">
  <itemref idref="cover.xhtml" properties="duokan-page-fullscreen"/>
  <itemref idref="Chapter1.xhtml"/>
```

图8-29　报错定位

在Sigil中可以使用快捷键「Ctrl+/」跳转到指定行，或依次点击「搜索→跳转到指定行」，然后再弹出对话框中输入要跳转的行号。

信息提示我们「duokan-page-fullscreen」不是ePub定义的属性。实际上这个是为了在多看阅读App中可以把当前页的图像以全屏的方式显示，属于阅读器自身的私有属性，并不是IDPF官方制定的规范。当你需要在多看阅读器中显示全屏封面效果时，可以不必修改这个报错。

现在我们来看第2、3、4、5条报错：

ERROR	CSS-008	/OEBPS/Styles/Stylesheet.css (line 854, col 28)	An error occurred while parsing the CSS: Token ';' not allowed here, expecting a property name.
ERROR	CSS-008	/OEBPS/Styles/Stylesheet.css (line 871, col 28)	An error occurred while parsing the CSS: Token ';' not allowed here, expecting a property name.
ERROR	CSS-008	/OEBPS/Styles/Stylesheet.css (line 903, col 28)	An error occurred while parsing the CSS: Token ';' not allowed here, expecting a property name.
ERROR	CSS-008	/OEBPS/Styles/Stylesheet.css (line 915, col 28)	An error occurred while parsing the CSS: Token ';' not allowed here, expecting a property name.

图8-30　详细验证信息

根据第3列的信息，在「Stylesheet.css」文件中定位报错位置。

```
853     -webkit-column-count: 3;
854     /* Safari and Chrome */;
855 }
```

图8-31　定位报错信息

仔细浏览CSS代码并根据报错的具体信息（解析CSS时发生错误），发现是多了「;」符号。处理办法是删除这个符号，然后保存ePub文件，再点击「EPUB-Checker」界面上的「validate　EPUB（验证EPUB）」按钮。可以发现这个报错没有出现，这表示错误已经被修复了。

我们再来看下错误代码为「RSC-005」的报错信息：

ERROR	RSC-005	/OEBPS/Text/Chapter2_2.xhtml (line 304, col 9)	Error while parsing file 'element "rb" not allowed here; expected text or element "a", "abbr", "area", "audio", "b", "bdi", "bdo", "br", "button", "canvas", "cite", "code", "command", "datalist", "del", "dfn", "em", "embed", "epub:switch", "i", "iframe", "img", "input", "ins", "kbd", "keygen", "label", "link", "map", "mark", "meta", "meter", "ns1:math", "ns2:svg", "object", "output", "progress", "q", "rp", "rt", "ruby", "s", "samp", "script", "select", "small", "span", "strong", "sub", "sup", "textarea", "time", "u", "var", "video" or "wbr" (with xmlns:ns1="http://www.w3.org/1998/Math/MathML" xmlns:ns2="http://www.w3.org/2000/svg")'.

图8-32　详细验证信息

报错信息显示在解析HTML文件时，rb元素不应该在ePub里出现。这表示`<rb>`标签不是一个ePub推荐使用的标签，根据第3列的定位信息，在正文查看上下文，在「注音」（Ruby Annotation）推荐规范中发现rb元素的作用是对注音的字符进行分组，这比较重要，所以可以不用删除。

后面还有不同的错误，这里总结如下：

OPF-027：非标准属性；

CSS-008：解析CSS时发生错误；

OPF-029：图像文件不符合「image/jpeg」媒体类型；

PKG-022：图像原始格式为png而非jpg（提出了上个错误的解决办法）；

RSC-012：根据toc.ncx中的目录链接，没有找到对应的id（重新生成目录不能解决的话，依次点击「工具→Epub3 Tools→Generate NCX from Nav」）；

RSC-007：在ePub中没有发现所链接的资源文件（添加引用的字体文件，可参考后文中的字体子集）；

RSC-005：不推荐使用的HTML标签；

CSS-017：不支持使用CSS中的定位属性（这与ePub的动态布局冲突，可参阅1.5）；

错误种类：

1. ACC：与内容可访问性相关的问题和使用信息
2. CHK：与处理ePubCheck自定义消息文件相关的错误
3. CSS：CSS的问题和用法
4. HTM：ePub内容文件的问题和用法
5. MED：媒体的问题和使用（音频和视频）
6. NAV：ePub导航文件
7. NCX：传统ePub 2 NCX导航/ toc文件
8. OPF：出版物OPF文件
9. PKG：ePub容器文件
10. RSC：包含在ePub中的资源文件（图像文件、css文件等）
11. SCP：包含在ePub文件中的脚本

默认情况下，ePubCheck的消息分为四种：

1. FATAL：阻止检查ePub文件本身的严重问题
2. ERROR：几乎肯定会阻止ePub成功渲染的一个问题
3. WARNING：这本书的某个方面不符合ePub规范，但这本书可能会被大多数阅读系统渲染
4. USAGE：记录使用特定ePub功能或标记模式的消息

8.3　Fontmin——字体子集化方案一

一本书不一定全部使用一个字体文件中的所有字型，所以我们需要通过软件提取当前书籍使用的字型，把未使用的字型剔除出去，这个过程称之为字体子集化。使用字体子集化方案可以控制电子书的体积，同时也丰富了版面。



图8-33　字体子集示例

常用的字体子集化工具，有Fontmin、Fontsub、EpubRepack，下面介绍其中两款软件的使用方法。

Fontmin这个软件有优点也有缺点，对于制作ePub来说。首先如果你想要子集化一个字体，需要先确定一段文字，然后才可以根据这些文字执行子集化操作。但是ePub中不同类型的字体有可能是分散在一本书中的不同地方，需要由制作人员自己去收集要子集的文字。例如楷体段落分散在书中几十处，那么光复制粘贴文本就需要不少时间。

所以这个软件的使用场景主要是个别几个文字或1、2段集中在一起的文字，而不是考虑不同阅读系统有不同的显示字体，但是制作者一定要显示自己选定的字体情况下。

这个限制既是缺点又是优点，可以很方便的子集字体，和下一节要介绍的FontRepack不同。

在官网中下载客户端，解压下载的文件，演示的电脑为Windows　64位的，所以下载的是「Fontmin-v0.2.0-win64.zip」版本。双击解压后的

文件或点击鼠标右键选择「打开」选项。软件初始界面如下：

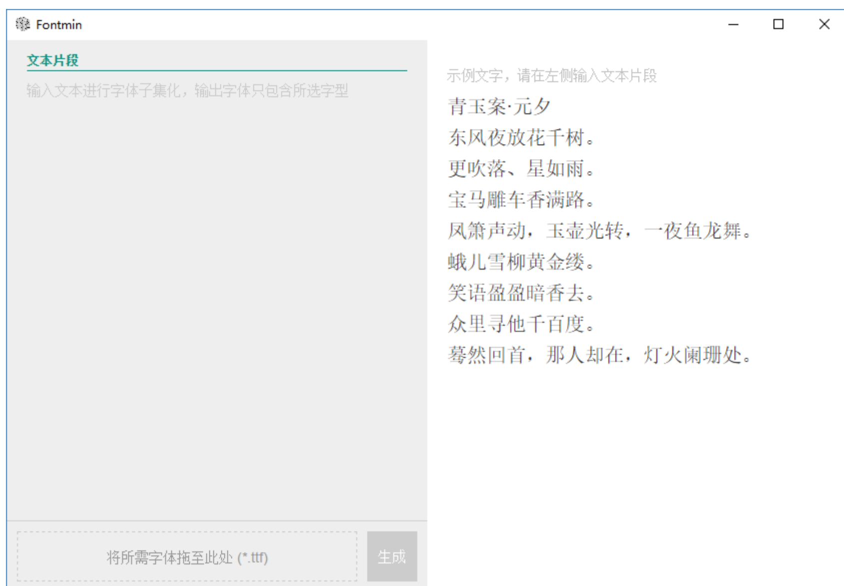


图8-34　Fontmin界面

首先复制粘贴需要子集的文字到「文本片段」输入框中，然后将要子集的字体添加到左下部。完成后如下：



图8-35　子集字体

点击「生成」按钮，等待软件编辑字体。然后弹出文件夹如下：

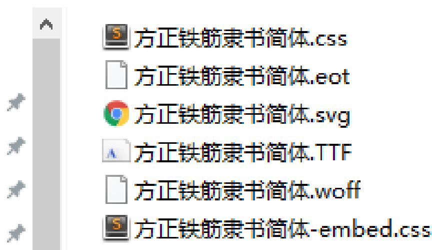


图8-36　子集后的配套文件

添加子集后的字体文件（TTF格式）到ePub中的Fonts文件夹下，修改文件名（不能是中文命名的）。HTML代码如下：

```
<p class="fontmin-demo">这是一沟绝望的死水，清风吹不起半点漪沦。......这是一沟绝望的死水，这里断不是美的所在，不如让给丑恶来开垦，看它造出个什么世界。</p>
```

CSS代码如下：

```
@font-face {
    font-family: "方正铁筋隶书";
    src: url("../Fonts/TieJinLi.ttf");
}
p.fontmin-demo {
    font-family: "方正铁筋隶书";
    font-size: 0.9em;
}
```

显示效果如下：

这是一沟绝望的死水，清风吹不起半点漪沦。不如多扔些破铜烂铁，爽性泼你的剩菜残羹。也许铜的要绿成翡翠，铁罐上锈出几瓣桃花；再让油腻织一层罗绮，霉菌给他蒸出些云霞。让死水酵成一沟绿酒，漂满了珍珠似的白沫；小珠们笑声变成大珠，又被偷酒的花蚊咬破。那么一沟绝望的死水，也就夸得上几分鲜明。如果青蛙耐不住寂寞，又算死水叫出了歌声。这是一沟绝望的死水，这里断不是美的所在，不如让给丑恶来开垦，看它造出个什么世界。

如果你的阅读系统支持自定义字体，上面的文字将会以「方正铁筋隶书」显示。

可以在资源管理器中查看子集后的字体大小，然后比对下原字体大小：

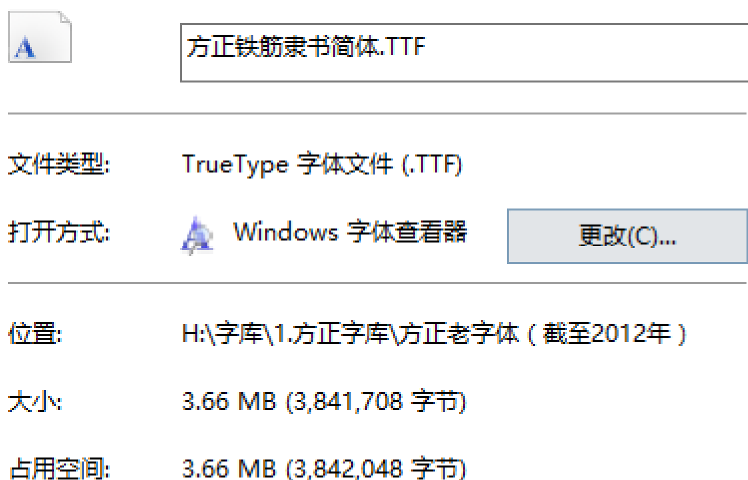


图8-38　子集前字体大小

因为我们只用了上面文本的几百个字型，其他没有采用的字型做了精简处理，所以子集后的字体相比原字体可以大大节省磁盘空间。

8.4 FontRepack——字体子集化方案二

FontRepack软件针对的是对整本书字体子集化，这种整本子集字体的好处是在一个阅读软件没有字体包的情况下，仍然可以保留多字体的排版效果。但由此带来的问题是，如果你的所有ePub都内嵌了子集字体，积少成多，也会大大浪费有限的磁盘空间。所以一方面推荐你选择一个经常使用的阅读器，这个阅读器最好带有丰富的字体，在制作时只需要在CSS中引用这些字体即可。

使用FontRepack软件来提取字体子集，该软件来自于Hi-PDA论坛的E-INK版块，FontRepack: epub多字体内嵌工具。

该软件的最新的版本是V1.40，这里就以这个版本做为演示。打开上面的帖子链接，到1楼的底部下载软件包——*fontrepack_v140.rar*。下载完成后，解压文件。下面是解压后的文件夹结构：

```
fontrepack_v140
│  config.xml
│  fontmap.ini
│  FontRepack.exe
│  README.txt
├─fonts
└─temp
```

打开README，开始按照教程配置软件。在fonts文件夹下新建一个文件夹，命名为「子集字体」，其他的名字也可以。复制`fontmap.ini`文件到子集字体文件夹下：



图8-39　fontrepack配置文件

使用记事本程序打开*fontmap.ini*文件，进行编辑。根据注释内容，进行

配置。fontmap初始配置：

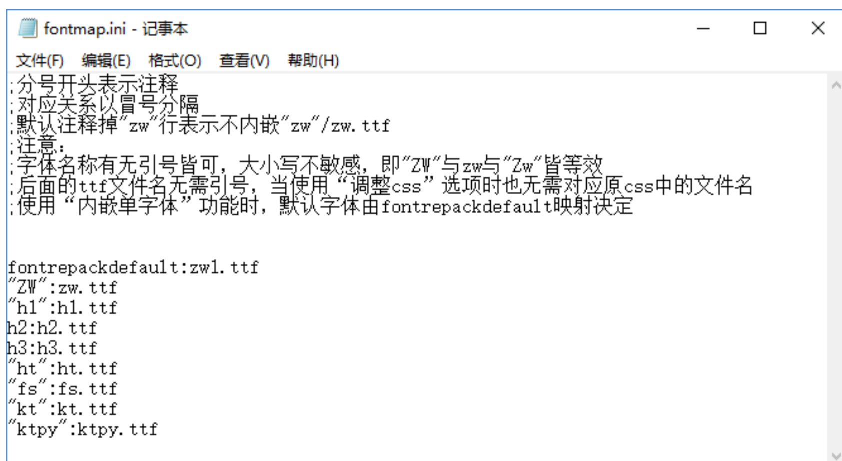


图8-40　fontmap.ini默认结构

修改文件内容，完成后如下：

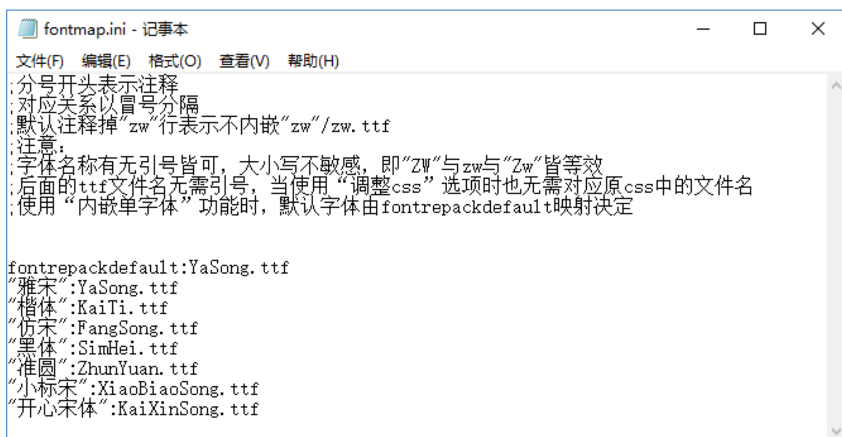


图8-41　修改fontmap.ini

完成上面的工作之后，我们需要在CSS中写引用外部字体样式。在Sigil中的「图书浏览器」的Styles文件夹下新建一个样式表，然后粘贴下面的代码：

```
/*方正书宋*/
@font-face {
    font-family: "宋体";
    src: url("../Fonts/YaSong.ttf");
}
/*方正仿宋*/
@font-face {
```

```
    font-family: "仿宋";
    src: url("../Fonts/FangSong.ttf");
}
/*方正准圆*/
@font-face {
    font-family: "准圆";
    src: url("../Fonts/ZhunYuan.ttf");
}
/*方正楷体*/
@font-face {
    font-family: "楷体";
    src: url("../Fonts/KaiTi.ttf");
}
/*方正黑体*/
@font-face {
    font-family: "黑体";
    src: url("../Fonts/SimHei.ttf");
}
/*方正小标宋*/
@font-face {
    font-family: "小标宋";
    src: url("../Fonts/XiaoBiaoSong.ttf");
}
/*开心宋体*/
@font-face {
    font-family: "开心宋体";
    src: url("../Fonts/KaiXinSong.ttf");
}
```

从上面的CSS代码中可以观察到，`font-family`的值对应fontmap中冒号前内容，URL所链接的字体文件对应fontmap中冒号后的内容。对应关系如下：

"开心宋体":KaiXinSong.ttf

font-family: "开心宋体"; src: url("../Fonts/KaiXinSong.ttf");

图8-42　对应关系

明白了上面的映射联系之后，我们再回到「子集字体」文件夹下。根据所链接的资源文件，在子集字体文件夹下添加字体文件，命名与fontmap中一致。完成后如下：

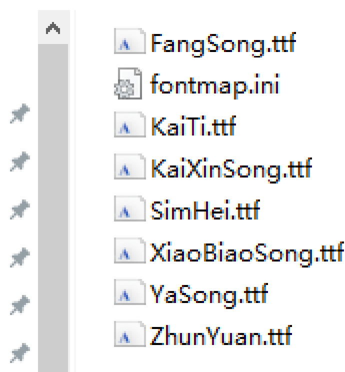


图8-43　字体文件命名

上面提到的字体，部分可以免费字体可以从方正字库官方网站获取，有两款需要购买（费用不高，建议用正版）：

- FangSong：方正仿宋_GBK（免费）
- KaiTi：方正楷体_GBK（免费）
- SimHei：方正黑体_GBK（免费）
- YaSong：方正书宋_GBK（免费，这里用免费的书宋代替）
- XiaoBiaoSong：方正小标宋_GBK（收费￥6）
- ZhunYuan：方正准圆_GBK（收费￥6）
- KaiXinSong：开心宋体2.1（免费）

方正购买后的字体只能是个人非商业使用，授权期限为永久。作为自己看书来说，已经够了。另外，制作的书建议不要分享出去，仅自娱自乐。

你可以根据自己现有的字体文件，按照上面的步骤一步步配置FontRepack。配置完成后的文件夹结构：

```
fontrepack_v140
│  config.xml
│  fontmap.ini
│  FontRepack.exe
│  README.txt
│
├─fonts
│  └─子集字体
│      fontmap.ini
│      FangSong.ttf
│      KaiTi.ttf
│      KaiXinSong.ttf
│      SimHei.ttf
│      XiaoBiaoSong.ttf
│      YaSong.ttf
│      ZhunYuan.ttf
│
```

```
└─temp
```

双击`FontRepack.exe`文件或点击鼠标右键选择「打开」，在「设置」中取消勾选「内嵌时调整CSS」：

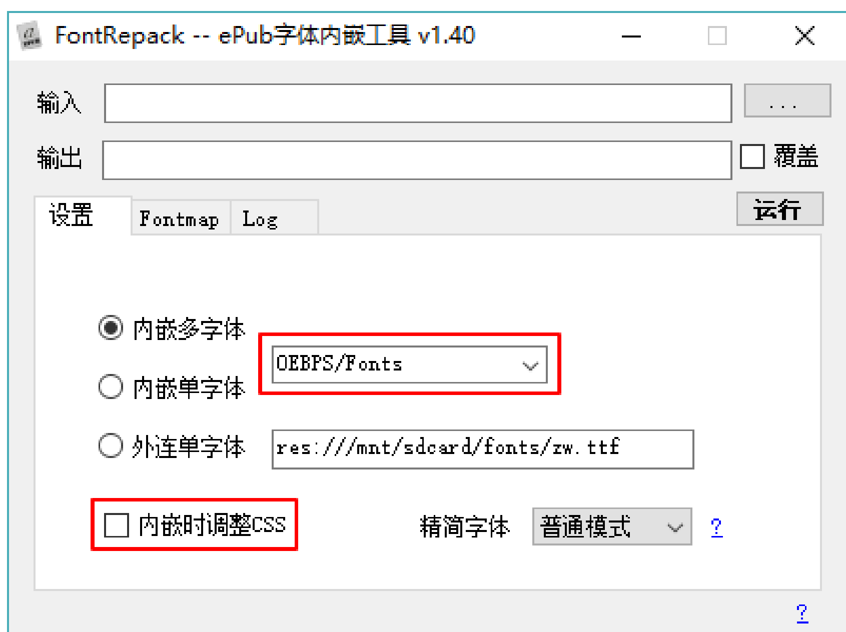


图8-44　FontRepack操作界面

精简字体的模式一般为「普通模式」，不用修改。内嵌多字体的文件路径如上设置。

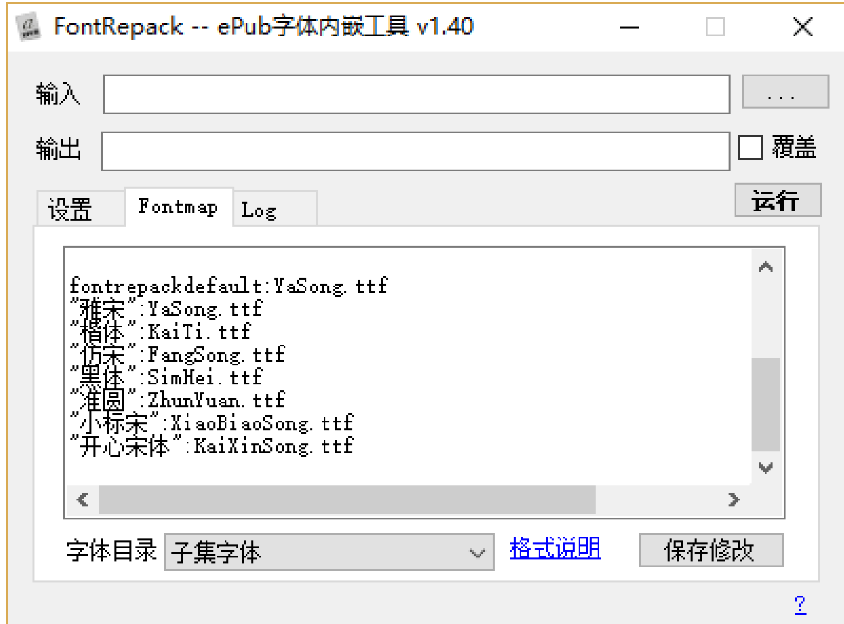


图8-45　配置文件预览

Fontmap中的字体目录选择配置的「子集字体」，这表示你可以在fonts文件夹中新建多个文件夹，按照不同的风格配置相关的字体。完成以上步骤之后，你可以开始给ePub子集字体了。

点击「输入」框右侧的按钮「...」，选择需要内嵌字体的ePub文件：

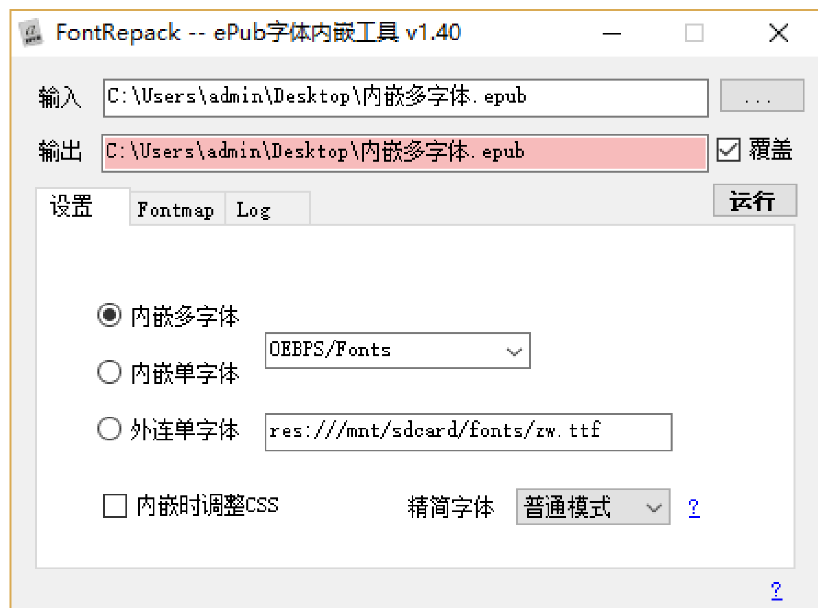


图8-46　载入需要子集字体的epub　图8-47　覆盖epub

「输出」地址默认是与源文件在同一文件夹下，当然你也可以修改输出的文件夹。在「输出」框的右侧有一个「覆盖」选项，当你勾选这个选项后，表示FontRepack不会新建一个epub文件，而是在输入文件的基础上进行修改。一般不推荐勾选「覆盖」！

然后点击「运行」按钮，运行时会在底部状态栏显示编辑信息：

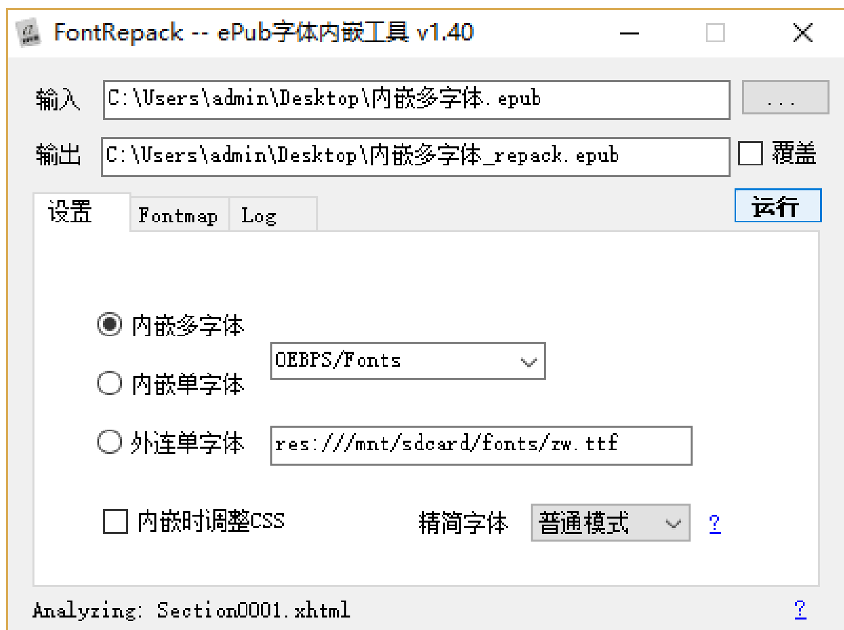


图8-48　正在子集字体...

如果提示缺字的话，一般不影响阅读。

缺字! ZhunYuan.ttf:99/150/202K　XiaoBiaoSong.ttf:112/215/215K

图8-49　完成后的提示

然后在阅读器中导入这个ePub文件，查看内嵌字体的显示效果。这里以多看阅读为例：

[No data]

《关雎》这首短小的诗篇，在中国文学史上占据着特殊的位置。它是《诗经》的第一篇，而《诗经》是中国文学最古老的典籍。当初编纂《诗经》的人，在诗篇的排列上是否有某种用意呢？这已不得而知。但至少后人的理解，并不认为《关雎》是随便排列在首位的。而汉儒的《毛诗序》又说："《风》之始也，所以风天下而正夫妇也。故用之乡人焉，用之邦国焉。"这里牵涉到中国古代的一种伦理思想：在古人看来，夫妇为人伦之始，天下一切道德的完善，都必须以夫妇之德为基础。

《关雎》的内容其实很单纯，是写一个"君子"对"淑女"的追求，写他得不到"淑女"时心里苦恼，翻来覆去睡不着觉；得到了"淑女"就很开心，叫人奏起音乐来庆贺，并以此让"淑女"快乐。另外，说它是情爱诗当然不错，但恐怕也不是一般的爱情诗。据我看来，这原来是一首婚礼上的歌曲，是男方家庭赞美新娘、祝颂婚姻美好的。

11:24　　1 / 1　本章还剩 1 页

图8-50　子集字体预览效果

这就完成一本书的字体子集，打开测试的ePub文件，在Fonts文件夹中增加了两个字体文件：

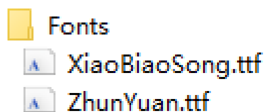


图8-51　子集字体文件

打开content.opf文件找到字体的资源链接：

```
<item id="easypubZhunYuan.ttf" href="Fonts/ZhunYuan.ttf" media-type="application/x-font-ttf"/>
<item id="easypubXiaoBiaoSong.ttf" href="Fonts/XiaoBiaoSong.ttf" media-type="application/x-font-ttf"/>
```

删除「easypub」。

8.5 subAllFont——字体子集化方案三

上节两个字体子集化方案，均需要依赖外部软件，且FontRepack软件操作有繁琐之处，配置起来略显麻烦。好友 @拐子 于2018年10月13日开发了一款Sigil插件——subAllFont，其功能正是字体子集化。

- 插件功能：直接将epub中已被引用的字体精简替换掉
- 插件地址：subAllFont
- 使用说明：请注意备份字体文件

参考8.6.1一节安装下载好的插件。

内容准备

插件使用前需要给HTML添加选择器，示例：

```
<p class="subFont">潡壿螲窲鴂鴮鶶蒐鷰駤髟鶟鵫鵂鷗鵢駥鵕鵜鳶鴻鳴駒鴑鵂鵗駐鴿鴂鳶鵨鵓鵂鵑鴰鶖鷐鵒鵟鵚鶏肆鳶鵝魌鵡鴎駭鷲鵃鳹鶩鵜鵏鵞鵤鵂鮚鴘鴰鵆鴣駮鶘鵩鷸</p>
<p class="subFont">鷺嗚鷇鸂鷉鷞鷭颮鷽鷟鷈鵔鶒鶆鷁鵿鷲鴿騠騧鷽鷄鵰鴷騃鷞鶤鵽鵯鵶鵨鶬鶬鷅鷗鵳鶦鸂鶘鶅鷟鶝鵯鷬鵅鷇鵞鸂鵪鶱鵬鵖鷺鶃鶔鵃鶢鶲鶋鷗鷗鴿鷉鷟騂鷭鵒鵬鶇鶡鶣鷭鷁</p>
```

在Styles文件夹中创建一个CSS样式表，并与HTML内容文档链接。样式表内容：

```
@font-face {
    font-family: "开心宋体";
    src: url("../Fonts/KaiXinSong.ttf");
}
.subFont {
    font-family: "开心宋体";
}
```

然后把「KaiXinSong.ttf」文件添加到ePub中。

插件处理

依次点击「插件→编辑→subAllFont」或插件按钮，弹出窗口如下：

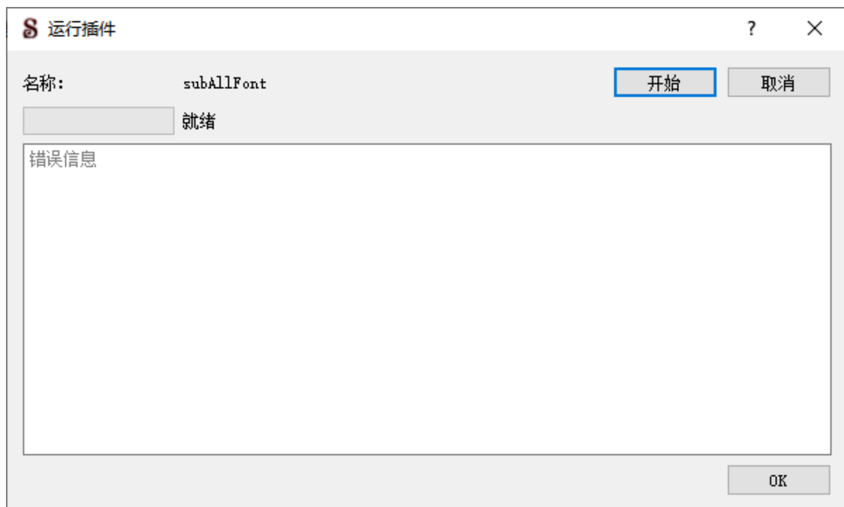


图8-52　点击「开始」按钮

点击「开始」按钮之后，弹出窗口如下：

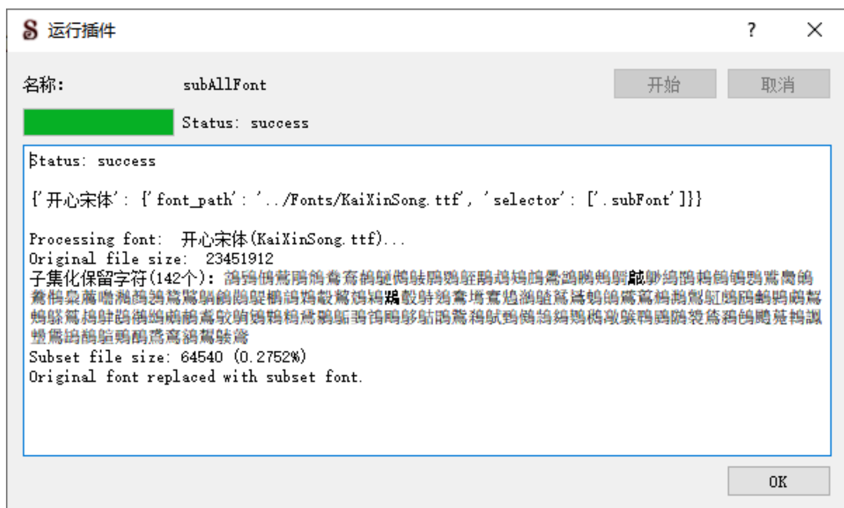


图8-53　插件运行结果

子集化后保存ePub文件，导入到多看阅读器中检查效果。无误后，即可发布。

如果你的字体子集化后在Sigil显示为子集化后的字体，在多看阅读中却显示为默认的黑体，这种情况可以使用FontLab、FontCreator等软件更改「字体嵌入性」属性。关于「字体嵌入性」解释见9.4节。

以上就是subAllFont插件的使用方法。

8.6 Sigil的常用插件

Sigil有不少可用的插件，可以帮助我们处理电子书制作的问题。比如DOCX文档转epub、AZW3文档转epub、epub转mobi、epub输出PDF等。本节从实际需求出发，对几个插件的使用做具体描述。

> 下载后的插件为zip格式，不需要解压，也不要变动文件名！

8.6.1 DOCX文档转ePub

DOCX格式是Microsoft Office Word创建的文件，这是一种常见的文件格式。Word本身也有另存为HTML文件的功能，但是另存出来的HTML文件冗余代码非常多，整理起来异常的耗费时间。所以一般不采用这种方式，把DOCX文档转成epub格式。该插件会通过连接Github的方式检查是否有更新，你也可以在GUI界面关闭「检查更新」。

- 插件功能：将DOCX格式文档导入Sigil，创建一个epub文件。
- 插件地址：DOCXImport
- 支持样式：标题、列表、表格、脚注和尾注、图片、粗体、斜体、下划线、删除线、上标和下标、链接、换行

安装插件

Sigil的「插件」功能在软件顶部的菜单栏中，如果你的软件没有「插件」这个菜单，请到官网（帮助→Sigil网站）更新你的Sigil版本。

在Sigil中依次点击「插件→插件管理」，这会弹出「首选项」窗口，如下：

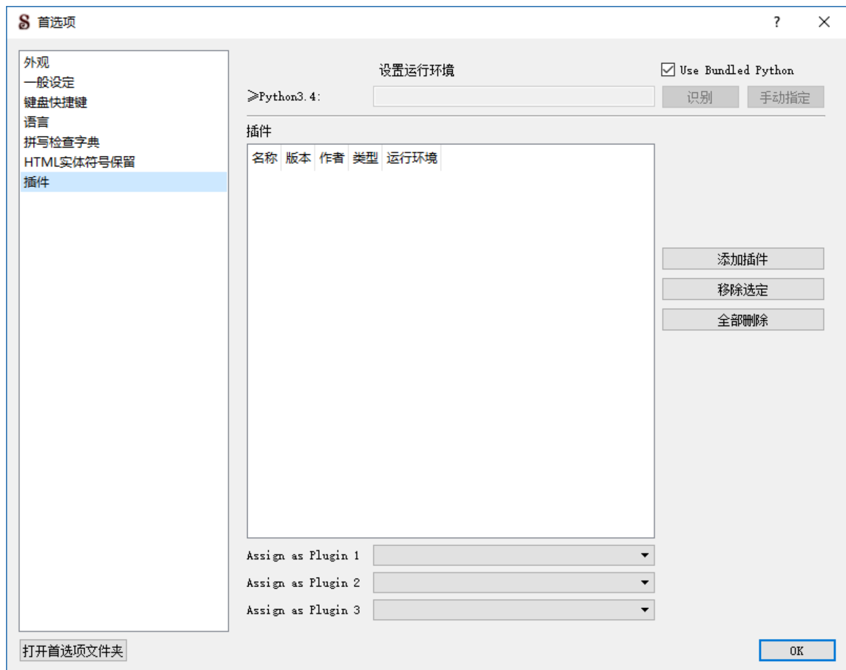


图8-54　「插件」首选项

点击右侧的「添加插件」按钮，在弹出对话框中选择下载好的「DOCXImport_v0.2.4.zip」文件，然后打开它。完成后会在插件列表中出现安装的插件：

名称	版本	作者	类型	运行环境
DOCXImport	0.2.4	DiapDealer	input	python3.4

图8-55　插件信息

想要卸载插件的话，先选择要卸载的插件，然后点击右侧的「移除选定」按钮。你也可以为工具栏快捷插件按钮分配3个快捷方式！

完成插件安装之后，点击右下角的「OK」按钮退出「首选项」窗口。

导入DOCX文档

你可以下载我提供的「sample.docx」文件，进行当前步骤的测试。这个文档包含一些常见格式，如标题、表格、注释、图片等。

依次点击「插件→输入→DOCXImport」，首先会弹出一个插件运行的界面，然后是插件的GUI界面。

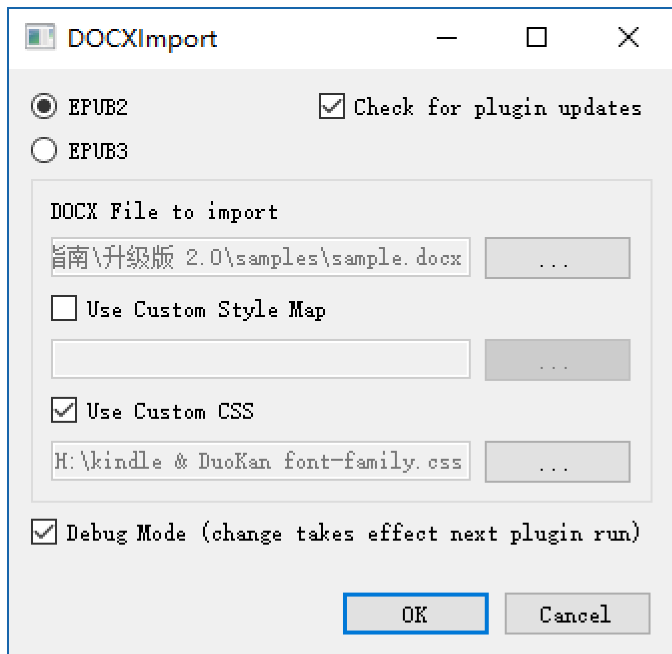


图8-56　DOCXImport操作界面

默认是创建ePub2标准的电子书，你也可以选择ePub3标准。然后是「Check for plugin updates」，上图是勾选状态，表示插件应该检查是否有更新，一般推荐勾选上。在「输入」框的右侧点击「...」按钮，在弹出对话框中选择要转换的DOCX文档，这里用的是「sample.docx」。完成后如上，导入的DOCX文档仍然会进行精加工，所以也可以在这一步把模板CSS文件添加进来。

然后点击底部的「OK」按钮，等待软件解析文件。完成后弹出下面的对话框：

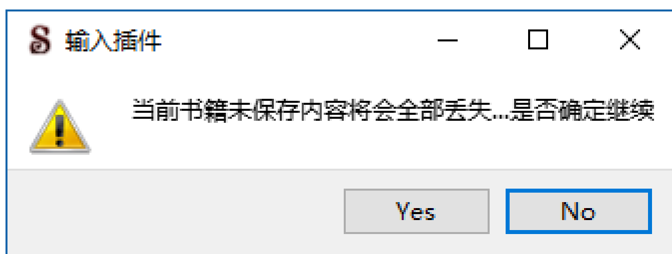


图8-57　输入插件提示

选择「Yes」按钮，然后点击「OK」退出插件运行窗口。

你可以查看经过转换后的epub文件，上面列举的样式都有保留，然后你可以在此基础上进行精加工。

8.6.2 ePub转Mobi

在上一步我们做了DOCX文档转epub的操作，这个epub文件还需要在做以下步骤才可以转成mobi格式：

- 添加封面（工具→添加封面，官方推荐尺寸2560×1600、350ppi）
- 创建目录（工具→目录→生成目录、工具→目录→创建HTML目录）
- 编辑元数据（工具→元数据编辑器，书名、语言、作者）
- 为默认第1次显示的HTML文件添加「文本」语义（添加语义→文本）

以上步骤可以参考第5章的教程部分，自己操作，这里不再描述。完成以上步骤后会在opf文件中的`<guide>`标签中出现如下：

```
<guide>
  <reference type="cover" title="封面" href="Text/cover.xhtml"/>
  <reference type="toc" title="Table of Content" href="Text/TOC.xhtml"/>
  <reference type="text" title="文本" href="Text/Section0001.xhtml"/>
</guide>
```

使用Sigil的「插入封面」功能会标记HTML文件为「封面」，插入图片为「封面图片」。标记为「文本」的页面会在打开时，出现在第一页。

- 插件功能：简单的kindlegen工具，输出azw3或mobi。
- 插件地址：kindlegen

下载要使用的插件，然后还需要下载亚马逊官方的KindleGen程序或者Kindle Previewer 3。如果你不想安装Kindle Previewer，可以只下载KindleGen，再根据下文的步骤，把这个文件放在指定文件夹中。

KindleGen是一款免费的电子书格式转换工具，需要使用命令行操作，也是亚马逊唯一官方支持的文件转换工具，它可通过它把HTML、XHTML或IDPF 2.0格式（带有XML.opf描述文件的HTML内容文件）的源文件创建为适合Kindle阅读的电子书格式。

不过你可能不习惯使用命令行操作，所幸这里有插件可以帮助你。

安装插件

参考8.6.1节内容安装下载的kindlegen插件，然后在「首选项」界面中点击左下角的「打开首选项文件夹」：

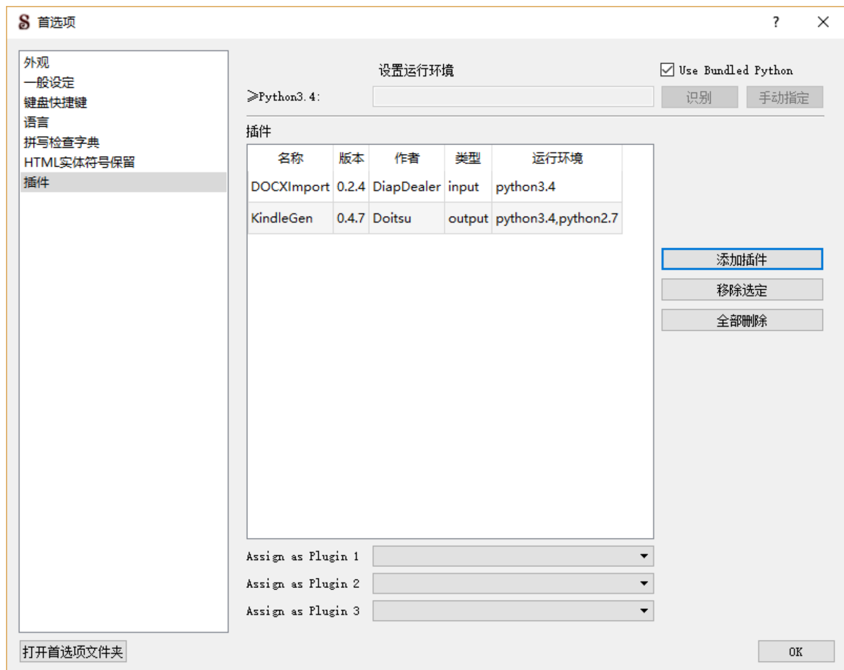


图8-58　「插件」首选项

在弹出的资源管理器中找到「KindleGen.json」文件，然后使用记事本程序打开它。参考地址：

```
C:\Users\admin\AppData\Local\sigil-ebook\sigil\plugins_prefs
\KindleGen
```

打开后的json文件如下：

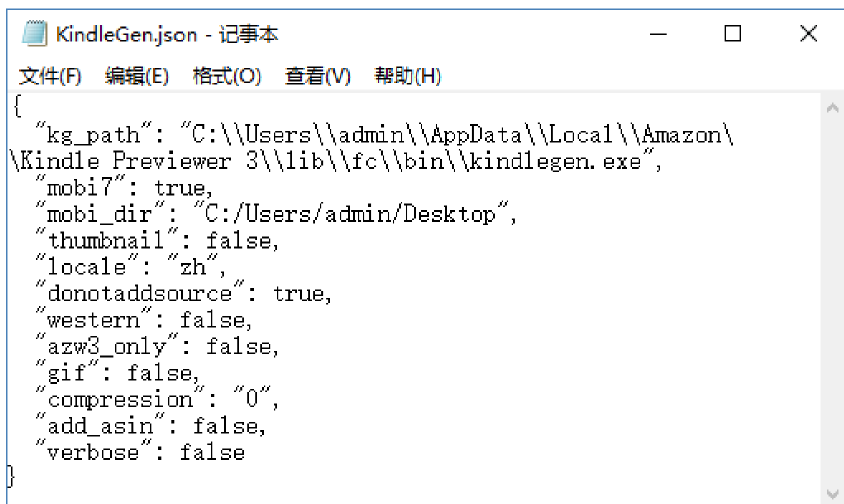


图8-59　KindleGen配置文件

根据「kg_path」值，依次创建文件夹，然后解压下载的「kindlegen_win32_v2_9.zip」，完成后如下：

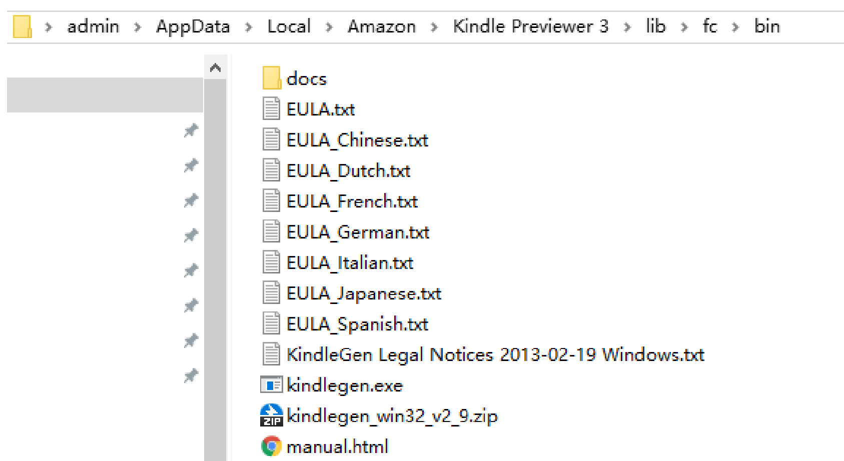


图8-60　kindlegen文件位置

地址如下：

```
C:\Users\admin\AppData\Local\Amazon\Kindle Previewer 3\lib\fc\bin
```

这是在你不想安装Kindle Previewer 3程序下需要做的工作。如果你安装了Kindle Previewer 3，创建文件夹及解压「kindlegen_win32_v2_9.zip」一步可忽略。直接进行下面的插件使用过程。

使用插件

依次点击「插件→输出→KindleGen」，会弹出插件运行的窗口，然后会弹出KindleGen的GUI界面：

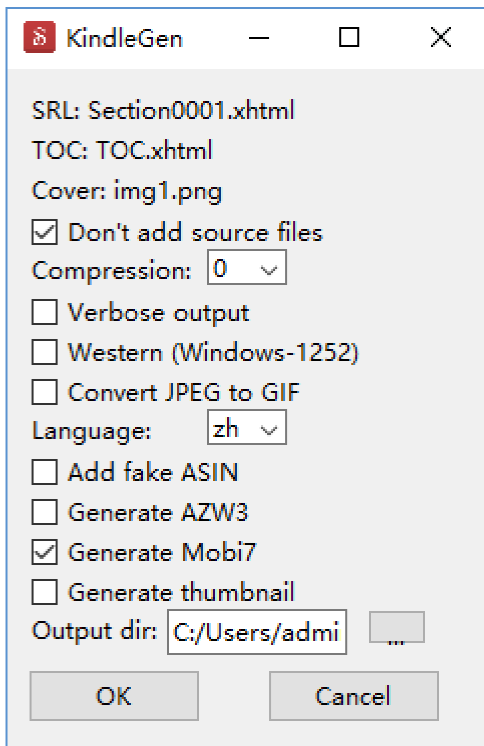


图8-61　KindleGen操作界面

设置如上，Output　dir是输出地址，你可以自定义输出地址。在上面的json文件中可以看到，默认输出地址是桌面。完成后如下：

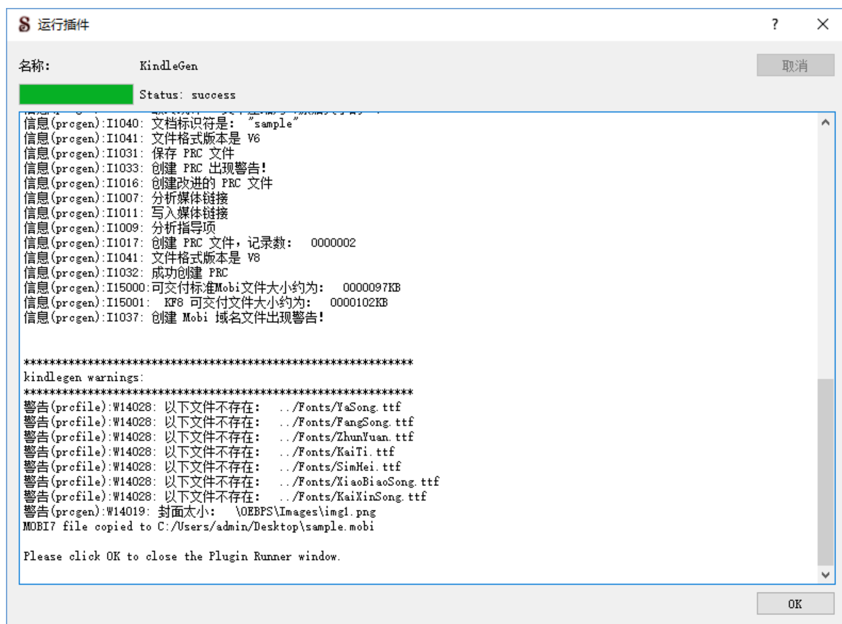


图8-62　输出mobi文件

「运行插件」窗口会在底部打印log警告信息，你可以根据警告信息，修改epub文件，然后再执行输出操作。

安装Kindle Previewer 3

如果你觉得创建文件夹麻烦，可以选择安装Kindle Previewer 3。根据上面的下载链接，下载安装包。双击安装包，执行软件的安装。在「安装地址」这一步选择默认地址。如下：

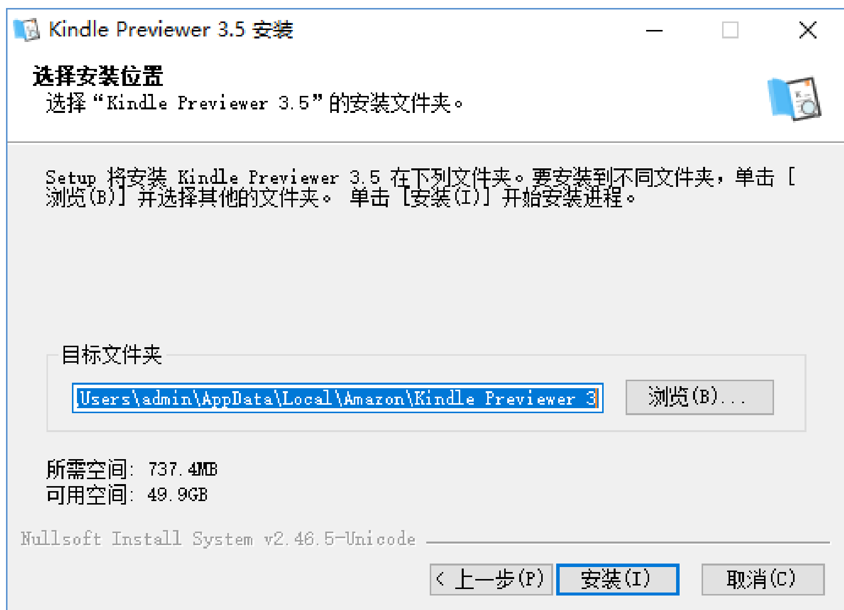


图8-63　安装Kindle Previewer 3

然后点击「安装」按钮，等待安装完成。安装完成后，查看软件的安装地址，发现kindlegen程序地址与json文件中的地址一致：

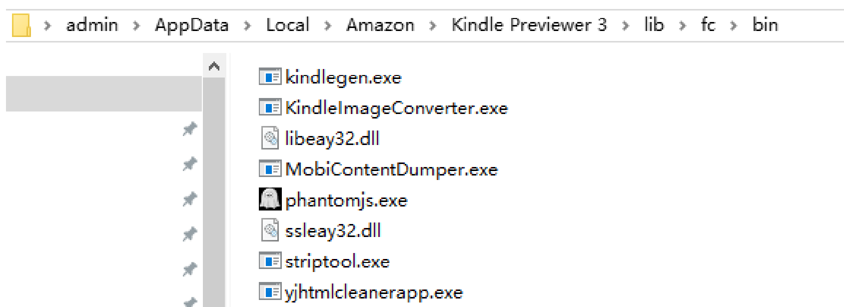


图8-64　kindlegen文件地址

软件安装完成后就可以在Sigil中执行输出操作了，步骤参考上文。

Kindle Previewer是一个模拟Kindle设备和应用程序显示书籍内容的使用界面工具。是亚马逊官方开发的一款kindle系列产品模拟器，可以模拟出kindle系列产品的功能。当然也可以用来阅读`.mobi`等专有格式的电子书，还可以对部分电子书格式进行格式的转换。

该软件默认是英文界面，可以设定为中文的，点击左上角的菜单图标，依次点击「View→Language→中文」，完成后界面就变成中文了。

比如你可以打开一本epub格式的电子书，使用它转成mobi格式。依次点击「文件→打开电子书」，在弹出对话框中选择要打开的epub电子书，等待软件转码。这个过程可能需要1～3分钟，视文件大小而定。

完成转码后，可以翻页预览。文件导出为mobi格式，则依次点击「文件→导出」，弹出窗口中选择文件保存位置，点击「保存」按钮。

以上就是epub转mobi的过程。转换后的mobi文件，可以使用Kindle Previewer 3打开预览：

SAMPLE

标 题测试：

标题1

标题2

有序列表测试：

1. 列表1
2. 列表2
3. 列表3

无序列表测试：

- 列表1
- 列表2
- 列表3

表格测试：

八王称号

王号	姓名	王号	姓名
汝南王	司马亮	楚王	司马玮
赵王	司马伦	齐王	司马冏
河间王	司马颙	成都王	司马颖
长沙王	司马乂	东海王	司马越

脚注与尾注测试：

晋朝（265年－420年）是中国历史的朝代之一，上承三国，下启南北朝，属于六朝之一。又可称司马晋[1]。晋朝又分为两个时期，即西晋（265年－316年）与东晋（317年－420年），合称"两晋"。西晋为晋武帝司马炎所建立，定都洛阳；东晋为晋元帝司马睿所建立，定都建康。两晋的政治体制为世族政治，政治制度由汉代的三公九卿制向隋唐的三省六部制过渡[2]。

图片测试：

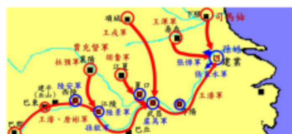


图8-65　Kindle Previewer 3预览截图

8.6.3　ePub2与ePub3互转

可以从ePub2标准升级到ePub3，也可以把ePub3降级为ePub2，这完全看你的需求。

ePub2转ePub3

- 插件功能：输出ePub3标准的文件。
- 插件地址：ePub3-itizer

仅对Sigil 0.8.6或更高版本提供了此插件的支持。该插件需要Sigil 0.8.2或更高版本才能运行。

在升级之前请先完成下面步骤：

确保您的epub2文件经过FlightCrew或EpubCheck验证（见8.2节）；

如果您的epub使用「.htm」或「.html」文件扩展名，则应首先将其更改为使用「.xhtml」文件扩展。修改步骤如下：

- 在Sigil的「图书浏览器」窗口中，选择所有.htm或.html文件；
- 右键单击您的选择，然后从上下文菜单中选择「重命名」；
- 在「重命名文件」对话框中，删除所有内容，并将其替换为「.xhtml」（无引号），然后点击「确定」。

参考上述安装插件的步骤安装下载的插件。然后执行如下操作：

a. 打开一本ePub2标准的ePub文件；
b. 依次点击「插件＞输出＞ePub3-itizer」；
c. 插件运行完成后，会弹出一个新文件保存窗口，重命名后选择文件夹保存。

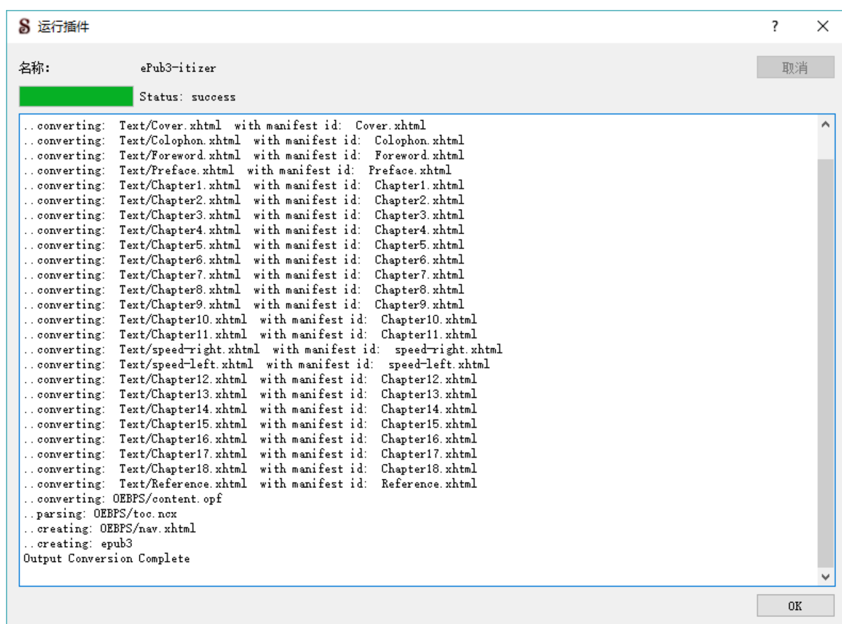


图8-66　ePub3-itizer运行窗口

升级好的ePub文件，最直观的体现是在Text文件夹下出现一个*nav.xhtml*导航文件，注意这个导航文件删除不了。

ePub3转ePub2

你也可以使用插件把ePub3标准的书降级为ePub2，不过这个插件并不是很完善，所以请小心使用！

- 插件功能：把大多数有效的ePub3小说书籍转换为有效的ePub2书籍，以便可以在较旧的ePub2应用程序和阅读器中使用。
- 插件地址：epub2 output

此插件不是一个成熟的epub3转换器，它只处理最常用的ePub3标签和

属性，并且不能转换包含JavaScript、MathML、音频、视频或SMIL文件等增强型ePub3书籍。输出后的ePub2文件，你需要再手动编辑某些文件。

该插件不附带GUI，但可以通过`ePub2.json`条目更改几个设置。json文件一般所在位置：

```
C:\Users\admin\AppData\Local\sigil-ebook\sigil\plugins_prefs\ePub2
```

使用记事本程序打开它，代码如下：

```
{
  "delete_epub_type": false,
  "ncx_from_nav": false,
  "output_dir": "C:\\Users\\admin\\Desktop"
}
```

delete_epub_type：如果您希望插件删除`epub:type`属性，请将此值更改为true。（默认情况下，它们将被更改为类属性。）

ncx_from_nav：如果您希望插件始终从*nav.xhtml*文件生成新的*toc.ncx*文件，请将此值更改为true。

output_dir：如果您不希望插件将ePub2文件写入到桌面上，请更改此路径。（请注意，在Windows系统中，您需要在路径名中使用两个反斜杠。）

8.6.4　使用EpubCheck插件验证ePub文件

在8.2节中我介绍了使用pagina来验证ePub文件，这里提供一个替代方式来验证你的ePub电子书。

验证ePub文件方法有好几种，在Sigil中你可以依次点击「工具→Well-Formed Check EPUB」验证你的ePub电子书，不过这个验证工具，判断错误的标准比较宽泛，换句话说，它不能检查出一些不规范的错误。所以我们需要专业的工具去验证我们的电子书，使之符合标准的规范。前面讲的pagina就是一个方式。

Sigil官方也提供了一个以插件的形式使用的专业验证工具——FlightCrew。

FlightCrew

该插件需要在Sigil 0.8.9或以上版本方可使用，有三种系统版本：

1. win：Windows 32位和Windows 64位
2. osx：Mac OSX 10.9或更高版本
3. unx：Linux 32位和64位

请根据你的系统下载适合的版本！

使用方法：依次点击「插件→有效性→FlightCrew」，根据检测出来的问题信息修改。见5.4.13节。

EpubCheck

- 插件功能：验证ePub电子书。
- 插件地址：EpubCheck

这个插件是一个非常简单的epubcheck 4.0.2封装插件，它允许您使用Sigil检查当前打开的书。

请注意，由于Epubcheck Java应用程序在后台执行，因此您必须等待几秒钟才能显示消息。建议在使用该插件验证ePub电子书前，先进行备份！虽然这个插件不太可能破坏你的书。

在安装插件之前，请确保满足以下系统要求：

1. Python 2.7.x（或Python 3.4）
2. Java 1.6或更高版本

Sigil 0.9.0及更高版本集成了Python 3程序，在使用pagina时我们已经安装Java 1.8环境，所以这里不再演示。如果您确定您的计算机上安装了Java，则会打开Windows资源管理器，并查找以下文件夹：

```
C:\Program Files\Java
C:\Program Files (x86)\Java
```

参考上文中安装插件的步骤，安装插件，这里不演示。

您可以通过`EpubCheck.json`文件控制三个EpubCheck插件设置。json文件一般所在位置：

```
C:\Users\admin\AppData\Local\sigil-ebook\sigil\plugins_prefs
\EpubCheck
```

代码：

```
{
  "clipboard_copy": false,
  "update_check": true,
  "java_lang": "en"
}
```

java_lang：强制Java使用不同的语言设置来运行`epubcheck.jar`。可以使用以下语言代码：

```
'en','de','es','fr','it','ja','nl'
```

请注意，日语输出不能在非日语Windows PC上工作。

clipboard_copy：自动将验证结果复制到系统剪贴板。（默认情况下禁用此设置，某些Linux/macOS版本可能无法使用。）

update_check：可用于禁用自动更新检查。

完成上述步骤之后，可以依次点击「插件→有效性→EpubCheck」来验证你的电子书。

然后根据验证结果，阅读报错原因，双击报告代码，定位错误，再决定是否进行修改。

8.6.5　搬运注释

把TXT格式的文本制作成ePub电子书，注释是一个比较大的问题。在10.6一节中介绍了不同阅读器对弹出注释的实现方式，无论是采用注释图片还是链接注释都需要把注释搬运到HTML文件的底部。如果每章的注释已经在HTML文件底部，只需要用正则表达式批量修改成某一规范就行。遇到下面这种情况，注释数量十几个还可以人工手动复制粘贴，如果几十甚至上百个且分散到不同的HTML页面内，再用人工的方式就不合理了。

> 焦湖庙祝[1]有柏枕，三十余年，枕后一小坼孔。县民汤林行贾，经庙祈福，祝曰：「君婚姻未？可就[2]枕坼边。」令林入坼内，见朱门，琼宫瑶台，胜于世。见赵太尉，为林婚，育子六人，四男二女，选林秘书郎，俄迁黄门郎[3]。林在枕中，永无思归之怀，遂遭违忤之事。祝令林出外间，遂见向枕，谓枕内历年载，而实俄忽之间矣。（出《幽明录》）
>
> 【注释】
>
> [1]庙祝：庙中管香火的人。
>
> [2]就：去。
>
> [3]秘书郎：掌典籍或起草文书的官。黄门郎：黄门，宫门。黄门郎即与王宫通事的官员。汉置，称黄门侍郎。
>
> 明月峡中有二溪东西流，宋顺帝[1]升平二年，溪人微生亮钓得一白鱼长三尺，投置舡[2]中，以草覆之。及归取烹，见一美女在草下，洁白端丽，年可十六七。自言：「高唐之女，偶化鱼游，为君所得。」亮问曰：「既为人，能为妻否？」女曰：「冥契使然[3]，何为不得。」其后三年为亮妻，忽曰：「数[4]已足矣，请归高唐。」亮曰：「何时复来？」答曰：「情不可忘者，有思复至。」其后一岁三四往来，不知所终。（出自《太平广记》卷四六九，原名《微生亮》）
>
> 【注释】

[1]宋顺帝：南朝宋顺帝刘准

[2]舡：「船」的异体字。

[3]冥契使然：幽冥的契合使我们这样。冥契，一般指幽冥中注定的婚姻契约。

[4]数：冥数，指冥契婚合的期限。

对于这种文本，有3个问题：1.注释需要重新编号，保证注释序号具有唯一性；2.把注释内容汇总到一起，放在HTML文件底部；3.文中注释编号与文下注释内容一一对应，可以互相跳转。

- 插件功能：搬运文中注释至HTML页面底部并给注释编号
- 插件地址：footnote

见8.5.1一节中的插件安装教程，安装 @王萌萌 编写的注释搬运插件。

依次点击「插件→编辑→footnote」，在弹出窗口中点击「开始」按钮，插件开始工作。如果注释数量都对的话，将会显示如下：

焦湖庙祝[1]有柏枕，三十余年，枕后一小坼孔。县民汤林行贾，经庙祈福，祝曰：「君婚姻未？可就[2]枕坼边。」令林入坼内，见朱门，琼宫瑶台，胜于世。见赵太尉，为林婚，育子六人，四男二女，选林秘书郎，俄迁黄门郎[3]。林在枕中，永无思归之怀，遂遭违忤之事。祝令林出外间，遂见向枕，谓枕内历年载，而实俄忽之间矣。（出《幽明录》）

明月峡中有二溪东西流，宋顺帝[4]升平二年，溪人微生亮钓得一白鱼长三尺，投置舡[5]中，以草覆之。及归取烹，见一美女在草下，洁白端丽，年可十六七。自言：「高唐之女，偶化鱼游，为君所得。」亮问曰：「既为人，能为妻否？」女曰：「冥契使然[6]，何为不得。」其后三年为亮妻，忽曰：「数[7]已足矣，请归高唐。」亮曰：「何时复来？」答曰：「情不可忘者，有思复至。」其后一岁三四往来，不知所终。（出自《太平广记》卷四六九，原名《微生亮》）

[1]庙祝：庙中管香火的人。

[2]就：去。

[3]秘书郎：掌典籍或起草文书的官。黄门郎：黄门，宫门。黄门郎即与王宫通事的官员。汉置，称黄门侍郎。

[4]宋顺帝：南朝宋顺帝刘准

[5]舡：「船」的异体字。

[6]冥契使然：幽冥的契合使我们这样。冥契，一般指幽冥中注定的婚姻契约。

[7]数：冥数，指冥契婚合的期限。

然后删去「【注释】」，使用正则表达式修改成某一种注释规范。这里以多看注释标准为例：

文中：

```
查找：<sup><a href="#footnote(\d+)" id="note(\d+)">\[(\d+)\]</a></sup>
替换：<a class="duokan-footnote" href="#footnote-\1"><img alt="" src=" ../Images/note.png"/></a>
```

文下：

```
查找：<p><a href="#note(\d+)" id="footnote(\d+)">\[(\d+)\]</a>(.*?)</p>
替换：<li class="duokan-footnote-item" id="footnote-\2"><p>\3</p></li>
———
查找：<aside>
替换：<ol class="duokan-footnote-content">
———
查找：</aside>
替换：</ol>
```

见10.6.3节，添加注释所用到的资源，如注释图标、样式信息。

下面列出注释搬运常见问题：

1. 使用正则表达式验证文中序号、文下注释是否数量一致，有缺少则排查修改
2. 文下注释内容的p标签不要添加样式，格式设置成：＜p＞[1]注释内容。＜/p＞
3. 如果注释编号是①②③……、〔1〕〔2〕〔3〕……、〔一〕〔二〕〔三〕……等，统一使用正则表达式或搜索模板修改成[1][2][3]……
4. 文中注释序号前不可以是这种：`</span>[1]`、`</i>[1]`……，运行插件前先查找处理，保证[1]等前是汉字字符
5. 如果注释内容是：`<p>[2][3]注释内容。</p>`，需手动拆分成：`<p>[2]注释内容。</p>`、`<p>[3]注释内容。</p>`
6. 如果注释内容是：`<p>[4]同上。</p>`，把上面的注释内容复制到「同上」这一条

8.6.6 段落中注释修改成跳转注释

这个插件，它可以处理段落中具有某种规律性的文本。比如一个段落中注释内容是采用「（）、【】、{}、[]……」包裹的文本，这种格式使用正则修改成插件可处理格式，然后用插件处理。

- 插件功能：提取段落中特定格式文本，修改成跳转注释
- 插件地址：Easy Endnote Insertion

比如有一种弹出注释，它是在文中插入一个注释图标，注释内容写在图标的alt属性中，见10.6.1节。代码：

```
<img alt="三桠纸的原料。" class="footnote" src="../Images/note.png"/>
```

那么我们就可以使用正则表达式匹配这种模式的文本，将它替换成插件可以处理的规律文本。这个插件识别行内注释的依据是`^[...]`。

```
查找：<img alt="(.*?)" class="footnote" src="../Images/note.png"/>
替换：^[\1]
```

替换时注意「查找&替换」面板中的选项不可以勾选「懒惰匹配」。

然后运行这个插件，即可以把这个注释内容搬运到当前HTML文件的`</body>`附近，并添加CSS样式。处理后的文中脚注：

```
<a class="footnoteRef" href="#fn1" id="fnref1">Note 1)</a>
```

文下注释内容：

```
<div class="footnote" id="fn1">
  <p class="fn"><b>Note 1</b></p>
  <p class="fn">三椏纸的原料。</p>
  <p class="fnrightalign"><a href="#fnref1">Back</a></p>
</div>
```

如果你不喜欢这个样式，可以再使用正则表达式批量修改，并添加自己的注释样式。

8.6.7 Sigil插件清单

你可以根据自己的需求从下列清单中选择合适插件来完成你的工作，部分插件的使用上文中已作介绍，这里就不再赘述。还有一些插件，经我自己判断没有使用必要就没有再翻译，有兴趣的可以自己看下清单。此次更新日期为2018年1月18日。

该插件清单来自mobileread论坛，感谢这些技术大佬！这里只提供插件的发布地址，打开网页可能比较慢，部分资源也会加载不出来，但不妨碍浏览。

官方插件

官方插件有两个，一个是FlightCrew，另一个是testplugin。其中，FlightCrew插件在8.5.4节已作介绍，这里就不再讲述；testplugin插件的作用是验证Python环境已准备好运行Sigil插件。一般来说，如果之前电脑中安装过Sigil的低版本，卸载干净之后重新安装，都可以直接运行Sigil插件。所以testplugin这个插件可有可无，但当你的Sigil运行插件出现问题时可以运行这个插件，检测一下。

- 插件功能：验证Python环境已准备好运行Sigil插件
- 插件地址：testplugin

为了帮助Linux打包者和其他从源代码构建Sigil的人，我们创建了一个简单的`testplugin_v014.zip`来测试可用的Python 3.4解释器环境设置。

1. 使用附件`testplugin_v014.zip`。

2. 插件的测试需要是Sigil启动时生成的空白epub模板。
3. 依次点击「插件→插件管理」菜单，并确保在Mac OS X和Windows上勾选了「Use Bundled Python」复选框。在Linux上，请正确设置系统Python 3解释器的路径。
4. 点击「添加插件」按钮，在弹出对话框中找到下载的`testplugin_v014.zip`然后选中打开它，接着点击「OK」按钮退出插件管理对话框。
5. 依次点击「插件→编辑→testplugin」菜单启动插件，点击「开始」按钮运行插件。
6. 检查插件输出窗口中丢失或损坏的测试结果。

它测试的全套python 3软件包、Hunspell SpellChecking、使用bs4 sigil gumbo适配器、Sigil的内部python库以及许多BookContainer基本接口函数。

版本v014需要Sigil 0.9.8或更高版本来运行。

第三方插件

丰富的第三方插件是Sigil使用者开发的，我们在制作epub中遇到需要批处理解决的问题，可以在这里找到可能的解决方案。

AddiBooksXML

在5.5节的常见问题中，提到过关于iBooks的字体问题。该插件即为此问题的解决方案，在使用插件之前需要再计算机中创建一个文本文档，命名为com.apple.ibooks.display-options并修改文件类型txt为xml，然后复制5.5节的xml代码到新建的文本文档中。完成后保存文件，记住文件保存的位置，后面需要用到。

- 插件功能：使用此插件将iBooks专有的「com.apple.ibooks.display-options.xml」文件添加到epub的META-INF文件夹中
- 插件地址：AddiBooksXML

注意：这个插件通过连接到github定期检查更新的版本。请参阅本文结束部分以了解更多详细信息（包括如何禁用检查）。

- Sigil：v0.9.0或更高版本
- Python：Python 3.4+（捆绑或外部）
- 操作系统：Windows/Linux/OS X

Linux用户必须确保Tk图形化的Python模块已经存在。在基于Debian-based flavors上，这可以通过「sudo apt-get install python3-tk」来完成。

依次点击「插件→编辑→AddiBooksXML」菜单，使用文件对话框选择计算机中的XML文件。在Sigil中无法编辑此XML文件，因此在添加之前请确保XML是正确/完整的。您可以随时在Sigil中将其删除，并在出现问题时再次添加它。

添加完成的XML文件出现在Sigil的「Misc」文件夹中，但放心，它实际上是进入META-INF文件夹。

可配置的首选项是：

- check_for_updates：一个布尔值（默认为true）控制插件是否检查更新。如果您不想接收新版本的通知，请更改为false。

配置文件查看方式：依次点击「插件→插件管理」，在打开的「首选项」界面左下角中，点击「打开首选项文件夹」按钮；进入到sigil的安装文件夹中，点击「plugins_prefs」文件夹，在plugins_prefs文件夹中有一个与插件同名的文件夹——AddiBooksXML，进入到该文件夹中，使用记事本打开`AddiBooksXML.json`文件，更改check_for_updates选项。

AddKindleMediaQueries

这个插件的作用是添加kindle的媒体查询样式表，并优化KF7和KF8设备的图像显示效果。

- 插件功能：针对KF7设备（以像素为单位）和KF8设备（以%为单位）优化epub文件中的图像显示
- 插件地址：AddKindleMediaQueries

要求：

- Sigil：v0.9.3或更高版本
- Python：Python 3.4+（捆绑或外部）
- 操作系统：Windows/Linux/OS X

插件安装过程见上文，这里不再赘述。如果没有特别说明，下文的插件安装均是一种方式。

介绍：

该插件是一个编辑插件，可以自动对您的epub中的所有图像条目进行双重格式化，还可以将相应的媒体查询添加到主样式表中，以便在KF7设备（以像素为单位）和KF8设备（按百分比）优化显示。在Kindle电子阅读器运行时，CSS中的媒体查询将根据您的电子书是在Kindle KF7设备还是KF8设备上运行，然后选择适当的图像行进行显示。

用户选项：

Add media queries to this stylesheet：这是一个下拉选项列表，允许您选择正确的样式表来添加媒体查询。您选择的样式表应该是设置所有文本文件和图像样式的主要样式表。

Type preferred base pixel width value：这个文本输入选项允许你输

入你自己喜欢的基本宽度值，像素为%计算。此值仅用于计算KF8设备图像的百分比宽度。计算如下所示：图像宽度在像素/630×100

630px值是默认的基本宽度选项（或默认的屏幕宽度选项）。因此，用户可以选择为所有电子书图像输入较大或较小的基本宽度值，或者用户可以使用默认值——630px。如果您不确定如何使用此选项，那么您应该使用默认值。

运行插件：

打开一本epub文件，然后运行该插件。

警告：

这个插件只会对普通html图像进行双格式化，它不会格式化SVG图像。

请注意，上述所有Kindle　KF8图像尺寸的百分比计算都不准确，而是最佳平均值或百分比宽度，因为我只使用了一个默认的基本宽度值（平均屏幕宽度值）对于所有的Kindle　KF8设备在我上面的百分比计算。这将不可避免地在许多不同的KF8设备和传统的KF8（屏幕宽度略有不同）上显示图像大小的差异，因为我只使用了一个单一的最佳平均屏幕宽度值进行所有计算。

但是如果你想要一个特定的KF8设备上的百分比图像是完全正确的，那么这是可能的。您只需在用户对话框中输入该KF8设备的适当屏幕宽度值（以像素为单位）作为您的首选基本宽度值选项，然后将为该特定KF8的设备所有显示图像计算精确的百分比值。

ConvertAbs2RelCSSValues

这个插件用来转换CSS的绝对单位为相对单位em，也可以更改margin和padding的缩写形式。

- 插件功能：将cm（厘米）、mm（毫米）、inch（in，英寸）、pica（1pc＝12pt）和point（pt，磅，1pt＝1/72in）值转换为CSS中相对的「em」值
- 插件地址：ConvertAbs2RelCSSValues

要求：

- Sigil：v0.9.3或更高版本
- Python：Python 3.4＋（捆绑或外部）
- 操作系统：Windows/Linux/OS X

介绍：

这是一个编辑插件，可以自动将所有绝对值（如cm、mm、inch、pica和point值）转换为epub样式表中的相对em值。根据用户的选择，这个

插件也将所有的「padding」或「margin」的简写形式转换成长形式（见用户选项）。这个插件只能用于标准的epub（不是epub 3）。

下面显示了「margin」的各种简写形式（与「padding」相同的格式）的示例：

1. margin: 0;
2. margin: 1.5pt 0;
3. margin: 1pt 0pt 1.5pt;
4. margin: 24pt 0pt 12pt 0pt;

上述简写将转换为长形式。如下所示：

1. margin-top: 2em;
2. margin-bottom: 1.0em;
3. margin-left: 1.5em;
4. margin-right: 0em;

转换为长格式的另一个好处是，像Nook电子书阅读器在解析「margin」和「padding」的简写形式上存在问题，长形式不会。这就是为什么使用长形式更安全的原因，因为任何电子书阅读器在解析长形式都不会有问题。

对话框：

这个对话框将询问用户是否想将「margin」和「padding」简写形式转换成长形式。如果选择「是」，那么简写形式将被转换成长形式。但如果选择「否」，那么所有的简写形式值将被转换为「em」值。

运行插件：

打开一本epub文件，然后运行该插件。

警告：

插件不会转换像素值和百分比值为em值。

插件不会把`margin: 0 auto;`转换成长形式。

cssRemoveUnusedSelectors

这个插件和Sigil的「删除未使用的样式」功能一样，但更加强大。该插件不光可以处理class，还能处理（几乎）所有类型的样式规则的选择器。

- 插件功能：删除未使用的样式
- 插件地址：cssRemoveUnusedSelectors

它使用cssutils解析样式表和lxml/cssselect，来检查CSS选择器是否与

xhtml文件中的至少一个元素匹配。（Cssutils、lxml和cssselect都捆绑在Sigil安装程序中）。

所有在xhtml文件中没有对应元素的CSS选择器都被提示，由用户决定是否删除。

如果CSS解析器遇到错误，会引发警告，用户可以选择继续或停止运行插件。无论如何，为了安全起见，导致错误的特定样式表将保持不变（cssutils实现了许多但不是全部的CSS3功能，例如@media规则嵌套在其他@media规则中）。

通过lxml/cssselect将CSS选择器转换为XPath，然后在xhtml文件中进行检查。对这些（包含「:hover」、「:active」、「:focus」、「:target」、「:visited」）选择器永远不会匹配任何东西。对于尚未实现的选择器也是如此（*:first-of-type，*:last-of-type，*:nth-of-type，*:nth-last-of-type，*:only-of-type——只有在指定元素类型的情况下才能工作）。参考文档。

从0.2.0版本开始，就有一个图形界面来设置关于「重新定制」样式表输出选项的首选项（缩进、在规则中的最后一个值之后的分号等）。

图形界面介绍：

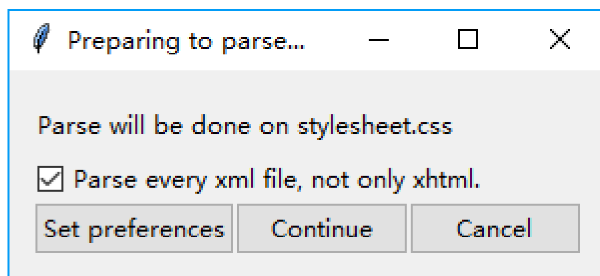


图8-67　解析CSS选择器

第一行的意思「将对stylesheet.css进行解析」，选项是「解析每个xml文件，不仅仅是xhtml。」点击「Set preferences」按钮，设置首选项。将弹出如下对话框：

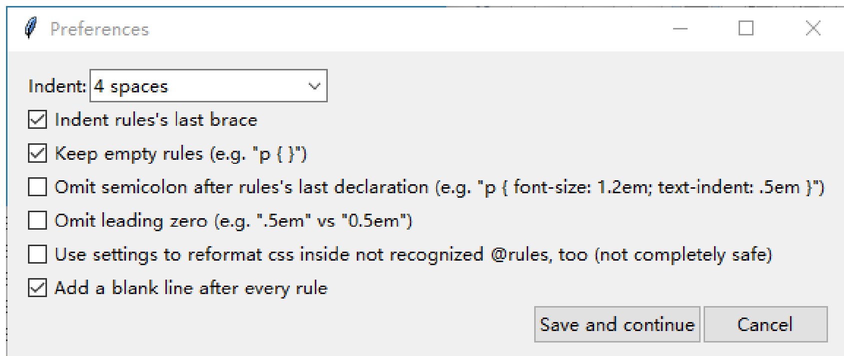


图8-68　设置首选项

第一行设置的是每条属性的缩进，可以选择Tab缩进或者是1-4个半角空格的缩进。下面分别介绍每个选项的意思：

- Indent rule's last brace：缩进样式的最后一个大括号（这里可以选择不缩进）
- Keep empty rules (e.g. "p {}")：保持空白规则（例如：「p {}」）
- Omit semicolon after rule's last declaration (e.g. "p {font-size: 1.2em; text-indent: .5em }")：忽略最后一个属性后的分号（例如：「p {font-size: 1.2em; text-indent: .5em }」）
- Omit leading zero (e.g. ".5em" vs "0.5em")：忽略小数点前的0（例如：「.5em」vs「0.5em」）
- Use setting to reformat css inside not recognized @rules, too (not completely safe)：使用设置来重新格式化内部不识别的@rules（不完全安全）
- Add a blank line after every rule：每条样式之后添加一个空行

设置完成以后，点击「Save continue」按钮保存并继续操作。或者点击「Cancel」按钮不保存并退出。然后切换到「Preparing to parse...」对话框点击「Continue」按钮继续操作。等待插件运行成功就行了。

RemoveInLineStyles

这个插件可以把内联样式提取出来，到一个外链样式表中。因为部分有的阅读器会特别注意此类问题。

- 插件功能：搜索并删除所有内联样式（xhtml标签中的样式属性），并将其替换为存储在外部样式表中的等效class
- 插件地址：RemoveInLineStyles

工作方式：

它将使用cssutils解析所有的样式表，原因是防止样式表文件名和class名称冲突，然后创建新的样式列表。

然后，将解析每一个xhtml文件，检查每个标签，如果存在style属性，将会删除它，查看当前新创建的规则列表，看看是否存在完全匹配成功

的，如果不添加新的类名就用新的规则。然后添加正确的类属性来替换已删除的样式属性。请注意，为了安全起见，不会触及svg或mathml元素/区域内使用的内联样式。

如果找到内联样式，那么新创建的外部样式表链接将在</head>标签之前插入到head元素中。

如果在xhtml文件中没有找到内联样式，将不会创建新的CSS样式表，除非在电子书中找到至少一个内联样式。

警告：

1. 请注意，head标签中style标签内的任何样式都将被忽略。
2. 不应该用于将CSS规则与内联样式混合的代码，因为内联样式具有更高的特异性，因此作者意图使用内联样式可能会取代先前的任何内容。

8.7 Prince——ePub转PDF

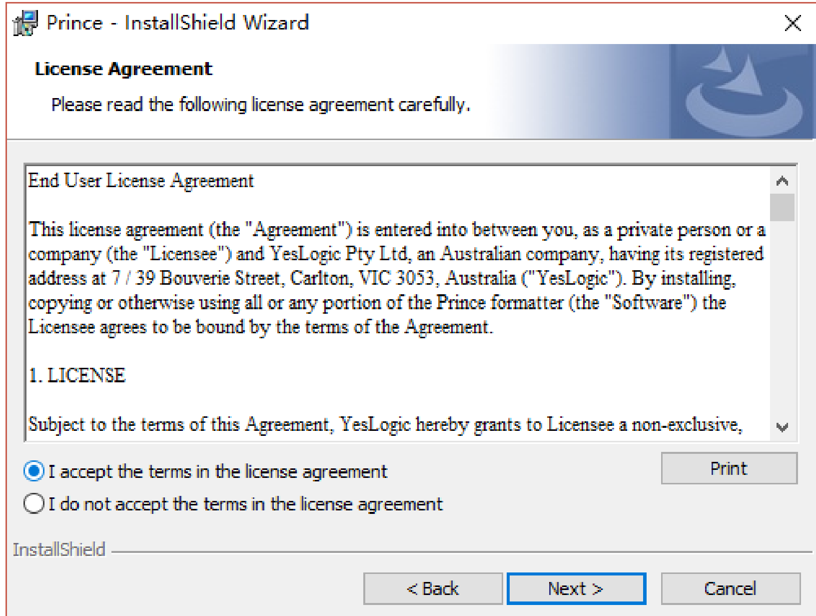
ePub文件转PDF文件，在Sigil中使用PrincePDF插件，与此插件配合使用的有一个打印程序。这个过程非常复杂，主要是用CSS来控制输出的PDF效果。本文将尽量详细的描述ePub电子书转PDF的过程。在操作前先完成插件和打印程序Prince的安装。

安装打印程序

插件功能：输出epub文件为PDF文件

插件地址：PrincePDF: PDF export

参考上文，安装下载的打印PDF插件。然后根据自己电脑系统版本下载对应版本的prince程序，演示使用的Windows10 64位的。这个打印程序是全平台支持的，所以不用担心使用不了。双击或点击鼠标右键选择「打开」prince-11.3-win64-setup.exe文件，依次点击「Next」：


End User License Agreement

This license agreement (the "Agreement") is entered into between you, as a private person or a company (the "Licensee") and YesLogic Pty Ltd, an Australian company, having its registered address at 7 / 39 Bouverie Street, Carlton, VIC 3053, Australia ("YesLogic"). By installing, copying or otherwise using all or any portion of the Prince formatter (the "Software") the Licensee agrees to be bound by the terms of the Agreement.

1. LICENSE

Subject to the terms of this Agreement, YesLogic hereby grants to Licensee a non-exclusive,

图8-69 同意用户协议

然后是选择安装地址，一般为默认地址，不用修改：

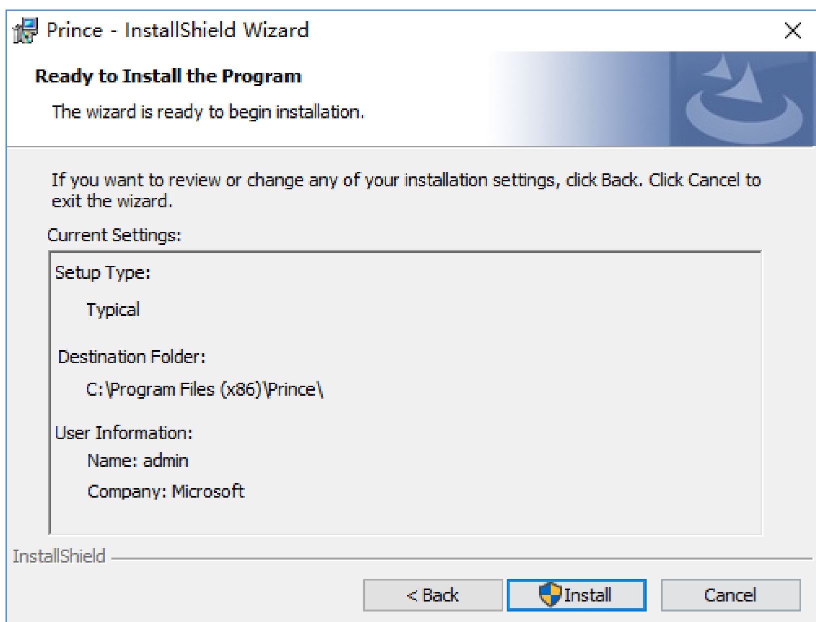


图8-72 执行安装程序

点击「Install」确认安装程序。完成后点击「Finish」结束安装。

依次点击「编辑→首选项→打开首选项文件夹」，在下面的地址中找到

「plugin.py」文件：

```
C:\Users\admin\AppData\Local\sigil-ebook\sigil\plugins\PrincePDF
```

使用记事本程序或EmEditor打开「plugin.py」文件，查看第29行的代码：

```
if iswindows :
    # the default path is: C:\Program Files (x86)\Prince\engine\bin\prince.exe
    prince_path = os.path.join(os.getenv('programfiles(x86)'), 'Prince'
```

这个打印程序地址需要与上文中安装的Prince程序地址一致，插件才可以运行。在桌面上选择Prince图标，点击鼠标右键，选择「打开文件所在的位置」，根据上面的地址信息确认：



图8-73　Prince软件安装位置

只要你完全是按照上述步骤操作的，就不需要再做任何修改。然后就可以跟着8.6.12一节输出你的PDF了。

8.7.1 页面布局

在你准备好内容之后，你需要使用CSS来格式化样式，让你的ePub更加有吸引力。比如多字体排版、如何格式化文本和段落、页面布局、背景、表格、插图等。

Web和PDF/Print的格式之间的主要区别是PDF被分页，这意味着内容放置在离散页面上，而不是一个连续的页面中，典型的连续页面是网页。

Prince允许您控制多个选项来影响PDF格式页面，从直接选项（如页面大小）到装饰、页眉和页脚、编号和分页选项等。

要定义字体或要在文档中使用字体，将使用`font-family`属性。Prince将尝试使用指定的字体，如果字型丢失，它将静默地回退到级联中的下一个字体。

解决上述问题的办法是根据8.3和8.4二节，对全书进行字体子集。为了保留排版效果，强烈建议你内嵌字体。

布局是在页面上设置文本和图片的方式。它定义了文档的最终外观。Prince通过CSS样式表确定页面的布局。

文本格式

文本样式可以通过CSS的文本属性来设置，具体内容见3.4.14节。

使用`font-family`属性为文本选择适当的字体后，您可以使用`font-size`属性指定不同的字体大小。

接下来你应该决定是否给它一些颜色。页面的默认文本颜色在body选择器中定义，但每个选择器都可以有自己的颜色。

```
body {
    font-family: "宋体", serif;
    color: #111;
}
```

特殊格式可以通过`font-style`、`font-weight`、`text-transform`或`text-decoration`属性实现。所有这些属性也可以用缩写属性`font`设置。

还可以使用`vertical-align`属性对内联框中的文本进行纵向对齐。

文本方向通过写作模式进行控制。

段落格式

现在您已经决定了文本的基本属性，您可以将注意力转向对段落的样式设置。`text-align`属性用于控制文本在其包含框中的对齐方式，而`text-indent`属性确定元素中第一行文本的缩进。如果`text-align`属性的值为left、right或者center，文本将分别对齐到左侧、右侧或中间。一般对于正文来说，需要设置的值时justify。在打印文本中，根据文本是否显示在左页或右页上对齐文本并不罕见。为了支持这个，在`text-align`属性中添加了两个新的值：inside与右页上的「left」和左页上的「right」相同，outside与左页上的「right」和右页上的「left」相同。

在中文中一般需要设置齐头尾规则，在CSS中有`text-align: justify;`和`text-justify`，但是在打印中，齐头尾规则并没有起作用。Prince添加了该属性`prince-text-justify`来定义如何处理CJK语言的对齐文本。prince-cjk值允许在CJK字符之间插入空格。

```
prince-text-justify: prince-cjk; /*定义CJK语言的文本对齐方式*/
```

格式化段落的另一个重要方面是文本行在打印页面上如何紧凑的显示。属性`line-height`可用于确定文本行的高度。

对于中英文排版的书，设置两端对齐后，还需要设置hyphens属性，以保证文本的对齐：

```
hyphens: none | manual | auto
```

当值时manual时，它会在单词之间以增加间距的方式排齐文本，换行的单词不添加连字符；auto的时候，单词间距固定，但在换行的单词中添加连字符-，保证文本的对齐。

Nikolai isky-orsako 18441908 was a ssian coposer a eer o a rop o proinent coposers known as he ie who worked toether in Saint etersr in the id-19th centry. Considered a aster o orchestration his est-known co-positions which inclde Capriccio spanol the ssian aster estial ertre and the syphonic site Scheheraade are con-sidered staples o the classical sic repertoire alon with sites and ecerpts ro soe o his 15 operas. Scheheraade is an eaple o isky-orsakos reent se o airy tale and olk s-ects he also let a considerale ody o oriinal ssian nation-alist copositions. is preparation o works y he ie or per-orance roht the into the actie classical repertoire althoh his editin o the sic o Modest Mssorsky created contro-ersy. As a shaper o a eneration o yoner coposers and sicians drin his decades as an edcator he is considered the ain architect o what the classical sic plic considers the ssian style o coposition.

图8.7.5 [illegible]

你可以根据需要选择一种值。

然后长链接（URL）的显示，建议设置为auto：

```
prince-linebreak-magic: none | auto;
```

即使在没有空格的情况下，也可以URL中的斜杠之后进行换行。对于想要精确控制断线行为的情况下，可以选择禁用它。

写作模式

写入模式描述脚本的方向性，即描述脚本的读取方向。在（X）HTML中，文档的语言由`lang`或`xml:lang`属性定义。为了控制文本的渲染，可以使用CSS的`direction`属性定义脚本的内联方向，即从左到右（例如拉丁语或Indic脚本）或从右到左（例如例如阿拉伯语和希伯来语脚本）。不过在ePub中禁止使用direction，所以我们需要另外想办法。

`writing-mode`属性描述了脚本的块方向，即文本是否应该水平布置，从顶部到底部（例如拉丁语或阿拉伯语），或垂直地从右到左（例如中文竖排）。默认值为horizontal-tb，即水平的从上到下。

当设置为竖排时，即从右到左、自上而下时，打印出来的PDF，每段占一页。目前还没找到解决的办法。

```
writing-mode: horizontal-tb | vertical-rl;
```

盒模型

`margin`描述的是内容区域的外边距，每个边距的大小可以通过属性`margin-top`、`margin-bottom`、`margin-left`和`margin-right`进行个性化设置，也可以使用缩写属性margin来一起指定所有四个边距。速记属性的语法是：

```
margin: top right bottom left;
```

如果只有：

- 三个值：左外边距等于右外边距；
- 两个值：左外边距等于右外边距，下外边距等于上外边距；
- 一个值：所有外边距都相等。

Prince扩展了个属性，`margin-inside`和`margin-outside`分别在内容的外部或内部定义边距。当设置`margin-inside`时，在左页上，内部位于右侧；在右页上，位于左侧。据这个属性，当你打印的PDF需要进行装订时，可以设置`margin-inside`定义书脊处的边距。

页面边距，你可以按照Word的默认边距进行设置，CSS代码如下：

```
@page {
    margin: 2.54cm 3.18cm; /*设置内容的外边距，设置上下左右的边距值。单独两个值的话，指的是上下2.54cm、左右3.18cm*/
    margin-outside: 3.50cm; /*定义书脊处的边距，用于装订。如果不需要可以直接删除*/
}
```

属性`padding`定义了内容模型中的填充。每个填充的大小可以通过属性`padding-top`、`padding-bottom`、`padding-left`和`padding-right`进行个性化设置，或者您可以使用缩写属性`padding`来一次指定所有四个属性。速记属性的语法是：

```
padding: top right bottom left;
```

解释：

- 三个值：左内边距等于右内边距；
- 两个值：左内边距等于右内边距，下内边距等于上内边距；

- 一个值：所有内边距都相等。

页面元素的背景可以使用属性`background-color`和`background-image`设置，或与速记属性`background`。可以使用`background`的标准属性进行设置。不过Prince使用`prince-background-image-resolution`属性扩展了背景控件，用于控制打印中的图像大小。这会在下文中说明。

表格

表格的设置可以参考本书中2.1.4和3.4.13两节。

不过，当一个表跨越多个页面时，`prince-caption-page`属性确定表格标题是否将显示在表格的第一页上，或仅在后续页面上显示，或者在表格的每个页面上重复显示。语法：

```
prince-caption-page: first | following | all;
```

不过在一个表格中试验这个属性，发现即使不设置也没有关系。前提是你的表格标题是在`<thead></thead>`中，并用`<th></th>`包裹。

多列

你可以根据3.4.10节设置你的文本以多列的方式显示。

默认情况下，列的高度是相等的，并且文本被分配到可用的列，以便每列中的内容的高度近似相等。在某些情况下，可能需要明确确定列的高度，并根据需要创建尽可能多的列。这是通过在多列块上设置`height`或`max-height`属性来实现的。语法：

```
height: auto | length | percent;
max-height: none | length | percent;
```

当内容在多列中排列时，Prince可以通过类似的方式确定列中断的位置，从而将内容分割成页面。`column-break-before`和`column-break-after`即是为此目的属性。语法：

```
column-break-before: auto | always;
column-break-after: auto | always;
```

浮动

当打印的文本包含图像时，文本通常被流动以包围这些图像。要使用CSS完成相同的操作，图像需要进行浮动——无论是文本左侧还是右侧，有时甚至是顶部或底部。`float`属性就是这样——它浮动一个元素，允许其他元素的内容流过它。

```
img {
    float: right;
}
```

属性clear可以被认为是float的姊妹属性。细节可见3.4.11节。

传统上，float在左右方向上移动。Prince扩展了这种行为，页面浮动在块方向上移动，指定一个元素应该浮动到顶部或底部，或者到最左边的列。

当设置float属性的值为top或者bottom时，将分别浮动到页面的顶部或底部。值column-top和column-bottom浮动元素，它会出现在列的顶部或底部，而column-top-corner与column-bottom-corner浮动元素到顶部或底部的最后一列。如果您想创建一个杂志风格的布局，将图像浮动到当前多列布局的右上角，这些可以很有用。

```
img {
    float: column-top-corner;
}
```

浮动元素可以在column-span属性的帮助下跨越多个列（参见列）。以下示例指示Prince使图像跨越两列：

```
img {
    float: column-top-corner;
    column-span: 2;
}
```

该值prince-snap指示Prince在多列布局的情况下将图像浮动到最接近的「结束」，即页面的顶部或底部或列的列。

```
img {
    float: prince-snap;
}
```

在印刷品中，通常必须处理左面和右面的页面，形成展开的效果。到放置元素时考虑到这一点，Prince扩展了float，属性值inside和outside分别移动所述元素，以扩展到内部或外部。inside在左页贴右；在右页贴左。

在这里你需要对图像进行裁剪，精确控制好尺寸和位置。

如果在多列布局中使用inside和outside值，则元素将浮动到其自然流中显示的列的内部或外部。

脚注

Prince支持使用该`float`属性的脚注。如果一个元素具有`float: footnote;`的属性，那么它将被浮动到页面的脚注区域中，并在在文本中会有一个序号标记。

每个脚注都会隐藏式地增加用于编号脚注的脚注计数器。必要时可以在每个新页面或章节重置脚注计数器。此示例将重置每个新页面上的计数器。

```
@page {
    counter-reset: footnote;
}
```

这个例子显示了一些简单的脚注，float-note类用于创建脚注。HTML代码：

```
<p>两晋的政治体制为世族政治，<span class="float-note">万绳楠（1994年）：《魏晋南北朝史论稿·第六章·世族统治下的西晋》. 第111页.</span>是隋唐的三省六部制的基础。<span class="float-note">邹纪万（1992年）：《中国通史·魏晋南北朝史·第二章 魏晋南北朝的社会型态》. 第101页 - 第106页.</span>而晋朝时期的农业、商业、手工业等相比较三国时期有了进一步发展。</p>
```

CSS代码：

```
span.float-note {
    float: footnote;
    font-size: .8em;
    color: #a52a2a;
}
```

打印的PDF效果：

两晋的政治体制为世族政治，[1]是隋唐的三省六部制的基础。[2]而晋朝时期的农业、商业、手工业等相比较三国时期有了进一步发展。

1. 万绳楠（1994年）:《魏晋南北朝史论稿・第六章・世族统治下的西晋》. 第111页.
2. 邹纪万（1992年）:《中国通史・魏晋南北朝史・第二章 魏晋南北朝的社会型态》. 第101页－第106页.

图8-76　脚注

脚注是文本中引用脚注的数字锚点。Prince将使用`::footnote-call`伪元素生成脚注。这是脚注调用的默认样式：

```
*::footnote-call {
    content: counter(footnote);
    font-size: 83%;
    vertical-align: super;
    line-height: none;
}
```

这将以比正文文本稍小的字体显示脚注计数器的当前值。该`line-height`声明确保脚注的上标位置不影响主文本的行高。

脚注风格可以自定义，比如使用不同的字体或颜色。

脚注放置在`@footnotes`页面的区域内，可以在`@page`规则中设置样式。请注意，如果页面上没有脚注，则脚注区域将不会显示在该页面上。代码：

```
@page {
    @footnotes {
        border-top: solid black thin;
        padding-top: 8pt;
        padding-left: 1em;
    }
}
```

显示如下：

1. 万绳楠（1994年）:《魏晋南北朝史论稿・第六章・世族统治下的西晋》. 第111页.
2. 邹纪万（1992年）:《中国通史・魏晋南北朝史・第二章 魏晋南北朝的社会型态》. 第101页－第106页.

图8-77　设置脚注样式

该规则在脚注区域的顶部添加一个边框和一些填充。

在多列布局，脚注也可以在每一列的底部，`float`属性通过使用值prince-column-footnote或prince-column-inline-footnote。

8.7.2 分页媒体

Prince输出PDF，这是分页媒体的一个突出例子。在为打印的分页媒体准备文档时，有一些差异至关重要：

分页

网页和PDF格式之间的主要区别是PDF被分页，即内容放置在分散的页面上。页面具有定义的大小并且可以使用预定义的Broken链接以特定模式布局内容。元素不仅可以左右浮动，还可以浮动到顶部和底部。可以选择和命名页面，对某些页面进行特定的处理。此外，了解分页是非常重要的：内容可能不适合页面，然后会溢出到下一页。

页面伸展

打印纸质媒介的基本单位是跨页。左页，名为Verso；右页，叫recto。它们大小是相同的，通常是对称的，彼此之间以书脊为中心。可以在recto或Verso中选择和命名页面，而Prince会使用inside和outside等扩展值，布局每个页面。

非交互式

最后但并非最不重要的是，打印的分页媒体本质上是非互动的。引用用户交互的所有CSS属性没有任何意义。

Prince允许您控制多个选项影响如何格式化页面，从直接选项，如页面大小，到装饰，页眉和页脚，编号和分页选项。

影响页面的CSS属性在页面规则中指定。例如，我们可以使用`@page`规则中的`size`属性来指定页面大小。

```
@page {
    size: A4;
}
```

在本节中，我们将显示可以在页面规则中指定的大多数属性，然后我们将显示如何控制分页，以及如何将页面样式应用于所选页面。

可以应用于页面的大多数属性都是现在将直接讨论的。其他属性需要了解选择页面，因此我们在该部分之后讨论它们。这些包括页面分页（分页符）和旋转宽的内容以适合页面，并使用`:blank`选择器。

高级纸质媒介的例子将显示页面媒体的高级用法，突出使用的命名页面，页面组，等等。

8.7.3 页面大小

可使用长度单位或由页面大小的关键字指定的size属性，也可以配合portrait（竖）、landscape（横）关键字进行组合。

```
/*A4纸*/
@page {
    size: A4;
}
/*A4纸横放*/
@page {
    size: A4 landscape;
}
/*上下、左右大小相等*/
@page {
    size: 30cm 40cm;
}
@page {
    size: 40cm 30cm landscape;
}
/*正方形纸张，宽度和高度相等*/
@page {
    size: 30cm;
}
```

以像素为单位指定页面尺寸也是可以的，例如创建演示幻灯片时。

```
@page {
    size: 1280px 960px;
}
```

Prince支持以下页面大小的关键字。（注意：关键字不区分大小写；例如「A4」和「a4」等价）。

表11　页面大小关键字及尺寸

Keyword	Page Size	Keyword	Page Size	Keyword	Page Size
A0	841mm × 1189mm	A1	594mm × 841mm	A2	420mm × 594mm
A3	297mm × 420mm	A4	210mm × 297mm	A5	148mm × 210mm
A6	105mm × 148mm	A7	74mm × 105mm	A8	52mm × 74mm
A9	37mm × 52mm	A10	26mm × 37mm	B0	1000mm × 1414mm
B1	707mm × 1000mm	B2	500mm × 707mm	B3	353mm × 500mm

B4	250mm × 353mm	B5	176mm × 250mm	B6	125mm × 176mm
B7	88mm × 125mm	B8	62mm × 88mm	B9	44mm × 62mm
B10	31mm × 44mm	C0	917mm × 1297mm	C1	648mm × 917mm
C2	458mm × 648mm	C3	324mm × 458mm	C4	229mm × 324mm
C5	162mm × 229mm	C6	114mm × 162mm	C7	81mm × 114mm
C8	57mm × 81mm	C9	40mm × 57mm	C10	28mm × 40mm

8.7.4 裁剪标记

排版杂志或传单时，可以直接打印到纸张的边缘。但是大多数打印机都会拒绝这样做。最好在更大的页面上打印，然后将纸张裁切成正确的尺寸。如果您使用marks属性，Prince将在您的内容周围创建出血和修剪区域（以及声明的页面大小）。

```
@page {
    size: A5;
    margin: 0;
    background: url("../Images/cover.jpg");
    background-position: center;
    background-size: 104%;
    background-repeat: no-repeat;
    marks: crop cross;
    prince-bleed: 1.5mm;
    prince-trim: 1.5mm;
}
```

新建一个空白HTML文件，在<style></style>粘贴上面的代码，然后把封面图片插入到ePub中。

本例使用一些我们上面介绍的特性，加上三个新的：marks、prince-bleed和prince-trim。这些选项可以进行裁剪和交叉标记，并指定出血和修剪区域。请注意，当我们添加这些选项时，内容的页面大小不会改变，该页面仍然是size属性指定的A5。相对于每个边缘（总共1个），增加了3mm的出血面积和3mm的修剪面积。Prince将打印纯色的背景颜色和重复的图片进入出血区域——我们将背景大小设置为104%，以便打印到出血区域。

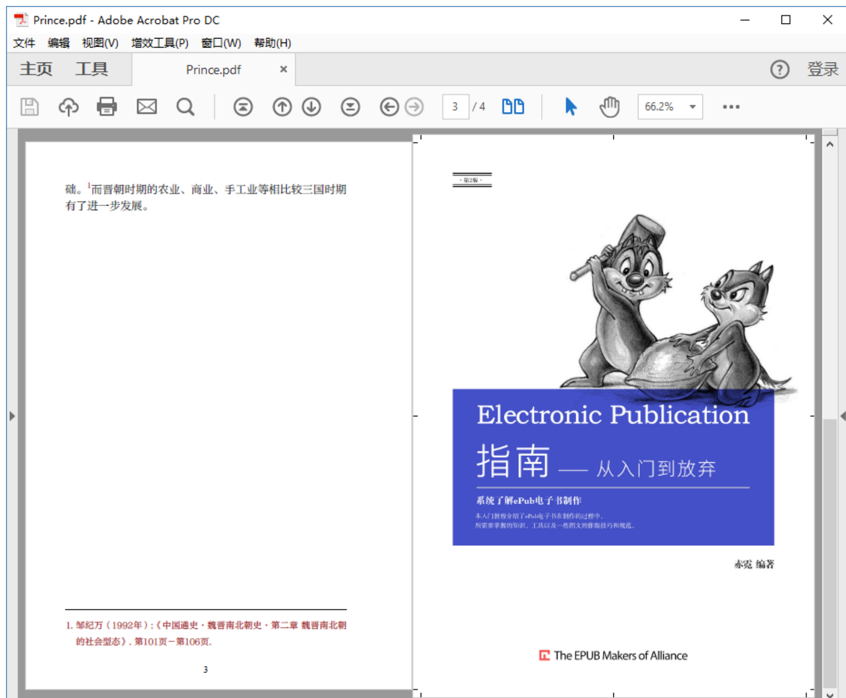


图8-78　裁剪标记

出血位指印刷时为保留画面有效内容预留出的方便裁切的部分。是一个常用的印刷术语，印刷中的出血是指加大产品外尺寸的图案，在修剪位加一些图案的延伸，专门给各生产工序在其工艺公差范围内使用，以避免裁切后的成品露白边或裁到内容。在做平面设计的时候我们就分为设计尺寸和成品尺寸，设计尺寸总是比成品尺寸大，大出来的边是要在印刷后裁切掉的，这个要印出来并裁切掉的部分就称为出血或出血位。

出血位的标准尺寸为3mm，就是沿实际尺寸加大3mm的边。

8.7.5　页面区域

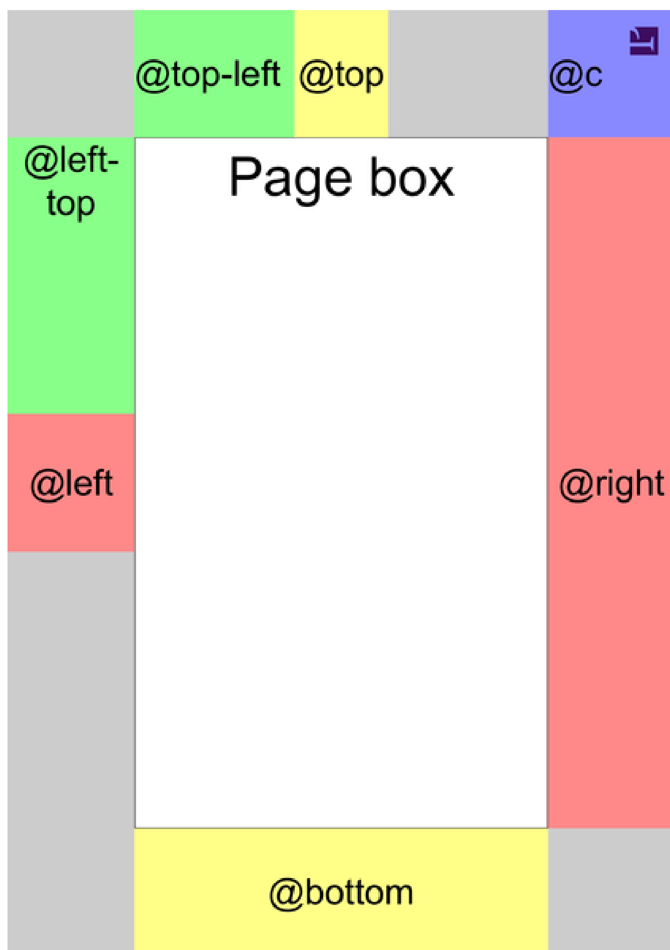


图8-79　Page各区域名称

大多数内容显示在一个或多个页面的页面框内。上图显示一些页面区域的位置。在这张图中，@page背景是灰色的，body的背景是白色的。前四个页面地区：`@top`、`@bottom`、`@left`和`@right`很容易理解，显示黄色和红色。这些页面区域中的每一个都有-center，例如`@top-center`，它和`@top`意义一样。您可以在页面区域中放置`content`属性。例如，您可以在每页底部打印当前页码：

```
@page {
    @bottom {
        content: counter(page)
    }
```

```
}
```

下面代码显示了一个简单的页脚demo：

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>小标题</title>
  <style type="text/css">
    @page {
      size: A5;
      @bottom { content: counter(page); }
    } /*在页面底部（页脚）插入页码*/
    @page:first {
      margin-top: 30%;
      @bottom { content: normal; } /*第一页不出现页码*/
    }
    body {
      font-family: serif;
    } /*内容区域为衬线字体*/
    h1,h2 {
      text-align: center;
    } /*文本水平居中*/
    p.content-first {
      page-break-before: always;
    } /*紧跟标题的第一段正文分页显示*/
  </style>
</head>
<body>
  <h1>chapter1</h1>
  <h2>subchapter</h2>
  <p class="content-first">In this example the @page rule specifies styles that apply to all pages: Then the @page:first rule overrides this for the first page only. It resets the content property for the footer and increases the top margin, printing the title of the novel in a reasonable place on the page (see Page style). This example also uses the page-break-before property to force a page break (see Page breaks).</p>
</body>
</html>
```

在此示例中，`@page`规则指定适用于所有页面的样式。然后`@page:first`规则将仅覆盖第一页的规则。它重置`content`页脚的属性，并增加顶部边距，在页面上合理的位置打印小说的标题。此示例还使用`page-break-before`属性添加强制分页符。

页面区域`@prince-overlay`是一个特殊区域，覆盖所有页面框。典型的用途是在所有页面上创建水印。代码：

```
@page {
    @prince-overlay {
        color: rgba(0,0,0,0.8);
        content: "Watermark";
        font-size: 20pt;
    }
}
```

上面的代码会在页面中央创建，显示内容为「Watermark」，你可以调整alpha值，以不同的透明度值显示。

页面区域`@footnotes`放置在页面框的底部，并包含脚注。你可以参考前文的脚注演示。

许多CSS属性可以应用于页面区域：

- 所有的margin、padding、border和background属性可以用来格式化页。
- vertical-align属性可以应用于任何页面区域以垂直对齐其内容。
- 当页面区域包含生成的内容时，可以使用许多内联样式属性，color和font可以将其应用于生成的内容样式。

页面区域的名称及默认属性：

表12　页面区域的名称及默认属性

名称	默认对齐		图中
	text-align	vertical-align	
@top	center	middle	yellow
@bottom	center	middle	yellow
@left	center	middle	red
@right	center	middle	red
@top-left	left	middle	green
@top-right	right	middle	not shown
@bottom-left	left	middle	not shown,
@bottom-right	right	middle	not shown
@left-top	center	top	green
@left-bottom	center	bottom	not shown
@right-top	center	top	not shown
@right-bottom	center	bottom	not shown
@top-left-corner	right	middle	not shown
@top-right-corner	left	middle	blue
@bottom-left-corner	right	middle	not shown
@bottom-right-corner	left	middle	not shown
@prince-overlay	left	middle	not shown
@footnotes	left	middle	not shown

页面区域中生成的内容

在上面的案例中我已经演示了一个page计数器，该page计数器是预定义

的，从1开始；它会自动在每个新页面上增量。

该page计数器可以使用复位`counter-reset`属性在正常流动非浮动元素内的任何块级元素中使用。这对于在文档的新部分重新启动页面编号非常有用。

下面以一个案例演示page计数器重启。HTML代码：

```html
<body>
  <div class="front">cover page, etc, ... </div>
  <div class="contents">Table of Contents, ... </div>
  <div class="body">
    <div class="chapter">
      chapter 1...
      <h1>First 1</h1>
    </div>
    <div class="chapter">chapter 2... </div>
  </div>
  <div class="appendix">appendix1</div>
  <div class="appendix">appendix2, ...</div>
</body>
```

CSS代码：

```css
.contents {
    display: block; /*以块状元素存在*/
    page: table-of-contents; /*给当前页编组，命名为「目录页」*/
    counter-reset: page 1; /*重启页码计数从1开始*/
}
@page table-of-contents {
    @top { content: "Table of Contents"; } /*在目录页顶部区域显示Table of Contents文本*/
    @bottom { content: counter(page, lower-alpha); } /*在页面底部以小写字母形式显示当前页页码*/
}
.body {
    display: block; /*以块状元素存在*/
    page: main; /*给当前页编组，命名为「主体」*/
    counter-reset: page 1; /*重启页码计数从1开始*/
}
@page main {
    @top-right { content: string(chapter-title); } /*在主体页的顶部右侧显示「章节标题」，你可以把它理解成变量*/
    @bottom { content: counter(page); } /*在主题页的底部显示页码*/
}
.chapter h1 {
    string-set: chapter-title content(); /*定义当前选择器为「章
```

```
节标题」，以便content可以调用*/
}
```

重新启动页面编号也使用了命名页面，我们将在下面更详细地解释。在上面的案例中我们创建了三种类型的页面：table-of-contents、main、未命名页面。

在某些文件中，特别是那些没有约束力的文档，例如office文档，可以显示每个页面上的总页数。可以使用pages计数器访问总页数。这是一个预定义的计数器，遍历在文档中的总页数。

```
@page {
    @bottom {
        content: "Page " counter(page) " of " counter(pages)
    }
}
```

这会在底部显示如：Page 1 of 4。

点击这里可以查看更完整的标题样式。

选择页面

通常需要将样式应用于某些页面，而不是其他页面。将它们应用到只有一些页面，或者将其应用到除了所选页面之外的每个页面。CSS和Prince提供了一些页面选择器，用于选择规则适用于哪些页面。

你可以参考本节开篇的案例——页脚demo。

在此示例中，`@page`规则指定适用于所有页面的样式，然后`@page:first`规则将仅覆盖第一页的规则。它重置`content`页脚的属性，并增加顶部边距，在页面上合理的位置打印小说的标题。此示例还使用`page-break-before`属性添加强制分页符。

当使用:first页面选择器选择每一章中的第一页时，可能需要添加`prince-page-group: start;`到每一章的第一个元素（如h1）中。

使用:left和:right页面选择器可以设置左右页的样式。这通常用于教科书中，将页码放在页面的外部顶角。

```
@page:left {
    @top-left { content: counter(page); } /*左页顶部左侧显示页码*/
    @top-right { content: string(book-title); } /*左页顶部右侧显示h1标题*/
    margin-left: 2cm; /*左页左外边距*/
    margin-right: 3cm; /*左页右外边距*/
```

```
}
@page:right {
    @top-left { content: string(chapter-title); }
    @top-right { content: counter(page); }
    margin-left: 3cm;
    margin-right: 2cm;
}
h1 { string-set: book-title content(); } /*标记为左页顶部左侧可以由content使用的文本*/
h2 { string-set: chapter-title content(); } /*标记为右页顶部右侧可以由content使用的文本*/
```

上面的代码，会将book-title即书名放在顶部左侧，chapter-title即章节名放在顶部右侧。书名从h1调用，章节名从h2处调用。同事也在左页的右侧、右页的左侧增加了外边距，这是为了装订的需要而设置的。

有时，在文档中选择第N页可能很有用。该页面选择器可用于执行此操作：:nth(N)

```
@page:nth(42) {
    ...
}
```

8.7.6 打印颜色

Prince支持RGB、RGBA、CMYK和命名专色。对于Prince的色彩管理，请参阅颜色管理部分。

RGB

Prince了解CSS基本颜色名称以及CSS Color Module Level 4中扩展颜色关键字列表。

要在CSS中使用RGB颜色，它可以表示为十六进制符号的值，作为`rgb()`函数或命名颜色。例如，红色可以通过三种方式指定：

```
color: #ff0000;
color: rgb(255,0,0);
color: red;
```

RGBA

RGBA颜色等效于RGB加上0和1之间的不透明度（或alpha）值。当不透明度为0时，颜色是透明的，当不透明度为1时，颜色与RGB颜色相同。RGBA颜色表示为一个`rgba()`函数。

```
color: rgba(255,0,0,1);     /*红色*/
color: rgba(255,0,0,0.5);   /*半透明的红色*/
color: rgba(255,0,0,0);     /*完全透明*/
```

CMYK

可以使用`cmyk()`或替代的等效`device-cmyk()`函数语法来指定CMYK颜色。可选的第五个值是Alpha通道，即不透明度。

```
color: cmyk(1,0,0,0);       /*青色*/
color: cmyk(0,1,1,0);       /*红色*/
color: cmyk(0,0,0,1);       /*黑色*/
color: cmyk(0,1,1,0,0.1);   /*透明红色*/
```

专色

Prince还支持可以用`@prince-color`规则定义的命名专色。还必须使用`alternate-color`属性指定其他颜色，使用RGB或CMYK颜色语法。这将在命名颜色不可用的情况下使用，例如在显示器上查看生成的PDF文件时。请注意，`alternate-color`不能是RGBA或CMYKA。

```
@prince-color MyColor {
    alternate-color: rgb(255,0,0);
}
```

专色可以与`prince-color()`0到1之间的指定色调值的函数一起使用，默认为1，或者以百分比表示。他们也可以启用套印：

```
color: prince-color(MyColor)                    /*tint 1.0*/
color: prince-color(MyColor, 0.5)               /*tint 0.5*/
color: prince-color(MyColor, overprint)         /*tint 1.0, overprint*/
color: prince-color(MyColor, 50%, overprint)    /*tint 50%, overprint*/
```

启用叠印的另一种方法是使用该`prince-overprint()`功能，允许指定不透明的颜色值。

8.7.7 色彩管理

色彩管理的目的是使颜色的外观保持一致，因为它们在不同的设备和媒体之间进行转换。这在将文件发送到打印机之前在计算机屏幕上准备PDF文件时具有优势——期望的结果将是使打印尽可能接近屏幕上的显示。完全等效是不可能的，也不仅仅是因为具有反射光的设备（例如纸张）和具有自发光的设备（例如屏幕）之间的差异，而是颜色管理有助于在感知中实现打印的紧密匹配。

颜色的转换通过成型设备来实现，将设备或颜色空间（即设备的颜色属性）表征的数据集保存为ICC颜色配置文件。颜色配置文件通常嵌入到图像文件中以表示图像的颜色空间。当颜色空间不知道时，颜色值基本上是任意数字，没有物理解释。

Prince的目标是在色彩管理的工作流程中，从输入文档和图像到PDF输出文件中保留颜色信息（即颜色配置文件）。

在Prince中，使用`--pdf-output-intent`命令行选项或使用`@prince-pdf`规则中的`prince-pdf-output-intent`属性来选择输出颜色配置文件。两者都将文件名或URL作为ICC配置文件的值。

```
@prince-pdf {
    prince-pdf-output-intent: url("ISOcoated_v2_eci.icc");
}
```

由于CSS定义了sRGB颜色空间中的RGB颜色，所以Prince将PDF格式的sRGB ICC配置文件标记为这些颜色。

```
@prince-pdf {
    prince-pdf-output-intent: url("sRGB.icc");
    prince-fallback-cmyk-profile: url("ISOcoated_v2_eci.icc");
}
```

除非有指示，Prince默认保留嵌入在位图图像中的ICC配置文件。如果图像没有嵌入式ICC配置文件，打印取决于图像的颜色模型。未标记的RGB图像将被假定为在sRGB颜色空间中。如果需要，CMYK图像将被假定为输出意图颜色空间（如果是CMYK），否则将返回CMYK配置文件的颜色空间。未标记的灰度图像将作为DeviceGray留在PDF中。

在印刷中，「浓黑」是CMYK墨水颜色的混合物，以产生比单独使用黑色墨水（「真黑」）更暗的色调。

该属性`prince-pdf-color-options`控制代表黑色和灰色的CSS　RGB值的编码，即红色，绿色和蓝色相等值的RGB颜色。

当使用该值时use-true-black，这种RGB值将被编码在PDF中的DeviceGray颜色空间中。CMYK打印机只能使用黑色墨水打印灰度级别。这是Prince的默认设置。

另一方面，该use-rich-black值指示Prince将所有RGB颜色作为RGB保留在PDF中。CMYK打印机应使用所有四种CMYK墨水的混合物打印RGB颜色。

```
@prince-pdf {
    prince-pdf-color-options: use-rich-black;
}
```

use-true-black当将CSS　RGB值转换为CMYK时，颜色转换将考虑该值。真正的黑色和灰色将被转换为CMYK，K通道中只有非零值。但是请注意，这不影响位图图像。

该属性`prince-pdf-page-colorspace`控制PDF文件中页面的颜色空间。它通过选择执行合成的颜色空间来影响透明内容到页面上的合成。Prince目前默认为RGB颜色空间。设置此属性可能有用，以避免在透明内容的颜色合成到页面之前转换颜色，可能导致扭曲的颜色。

8.7.8 打印图像

Prince支持JPEG、PNG、TIFF和GIF图像。

除非通过使用`width`和`height`属性来指定图像的显式大小，否则Prince将根据图像分辨率（DPI）确定内在大小，该尺寸可以使用以下`prince-image-resolution`属性进行覆盖 ：

```
prince-image-resolution: 300dpi;         /* set an explicit DPI */
prince-image-resolution: normal;         /* 1 image pixel maps to 1px unit */
prince-image-resolution: auto, normal;   /* auto-detect, fallback to normal */
prince-image-resolution: auto, 96dpi;    /* auto-detect, fallback to 96dpi */
```

并非所有图像都具有内部分辨率设置，因此有时需要指定明确的分辨率。或者指定分辨率normal，这意味着100像素宽的图像将在页面上与100像素宽的块相同。

该属性`prince-background-image-resolution`对于背景图像也是一样的。语法：

```
prince-background-image-resolution: dpi | normal | auto [ , normal | dpi ]
```

请注意，指定`prince-image-resolution`和`prince-background-image-resolution`仅影响图像的默认DPI。使它们在物理页面上更大或更小，它不影响图像中的像素数，因此PDF文件大小将相同。

为了减少PDF文件大小，JPEG图像可以在较低质量级别进行重新压缩，或将PNG图像转换为JPEG，并附带该`prince-image-magic`属性。语法：

```
prince-image-magic: none | snap-to-integer-coords | ignore-icc-profile | jpeg-verbatim | convert-to-jpeg( percent ) | recompress-jpeg( percent )
```

8.7.9 生成内容

生成的内容是在原始输入文档中找不到的文本和其他内容，但是使用`content`属性将其添加到样式表的输出中。

页眉和页脚

该`content`属性可以应用于页面区域，例如`@top`和`@bottom`，以创建具有生成内容的页眉和页脚。有关详细信息，请参阅页面区域。

```
@page {
    @top { content: "书名"; }
    @bottom { content: counter(page); }
}
```

列表标记

该`content`属性可以应用于::marker 伪元素以指定列表项的自定义标记。

```
li::marker { content: "No. " counter(list-item); }
```

列表项主要有两个部分:marker和content。

该content部分被渲染在列表项的边框内，并且与普通块（例如div或p元素）的方式相同。

该marker定位是通过所确定的`list-style-position`属性，并使用是风格::marker伪元素。

标记位置

默认情况下，列表标记在左边距区域的列表项之外。如果CSS属性`list-style-position`具有值inside，则标记将呈现为列表项中的第一个内联框。

```
ol {
    padding-left: 1cm;
    border: solid 2px darkgreen;
}
li { border: solid 1px black; }
li.inside { list-style-position: inside; }
li.outside { list-style-position: outside; }
```

此列表项设置`list-style-position`为inside，所以列表标记在主框内呈现

li。

此列表项设置`list-style-position`为outside，因此列表标记在主框之外呈现li。

当标记位置是outside，它被渲染在li主框之外。如果您希望标记出现在ol主框内，请确保给予ol足够的左填充或li足够的左边距。

标记类型

可以通过将CSS属性设置`list-style-type`为不同的值来选择不同的列表标记类型：

```
disc| hyphen | decimal | lower-alpha | ...
```

关于标记类型可见3.4.7节。

标记样式

列表标记也可以通过使用`list-style-image`属性替换为图像。

简写属性`list-style`可一次设置三个属性`list-style-image`、`list-style-position`和`list-style-type`。

或者，::marker伪元素可以用于对列表项标记进行样式化，从而完全控制其内容、宽度、对齐方式等。

```
li::marker { width: 2.4cm; }
li.left::marker { text-align: left; }
li.center::marker { text-align: center; }
li.right::marker { text-align: right; }
li.text_marker::marker {
    content: "Number " counter(list-item);
}
li.image_marker::marker {
    content: url("../Images/prince.jpg");
}
```

上面的代码显示为：

1. The marker of this list item is left aligned in the marker box.

2. The marker of this list item is center aligned in the marker box.

3. The marker of this list item is right aligned in the marker box.

Number 4 The marker of this list item has text "Number" in front of the list counter.

The marker of this list item is an image.

图8-80　列表样式位置

脚注标记

该`content`属性可以被应用到::footnote-call和::footnote-marker伪元素指定脚注自定义标记。

```
span.float-note::footnote-call {
    content: "#" counter(footnote);
    font-size: inherit;
    vertical-align: super;
}
```

这个规则将产生脚注，括号中的脚注数量如下：#1、#2、#3、...

Prince自动生成脚注标记，数字标记放置在脚注文本的前面。脚注标记类似于添加到列表项的标记（参见列表）。在大多数方面，并且可以使用::footnote-marker伪元素以类似的方式进行样式化：

```
*::footnote-marker {
    font-weight: bold;
}
```

此规则将生成具有粗体字体的脚注标记。

默认情况下，脚注标记在左边距区域的脚注之外。如果CSS属性`footnote-style-position`具有值inside，则标记将被渲染为脚注中的第一个内联框。

```
footnote-style-position: outside | inside
```

伪元素before和after

该`content`属性可以被施加到::before与::after伪元素之前或元素后添加生成的内容。例如，在标题前添加区号，或在文本块周围添加引号。

```
h1::before, h1::after { content: "###"; }
```

这将在h1元素之前和之后放置三个#号。

计数器和编号

计数器是CSS提供的执行编号的机制。它们可用于编号列表项、页面、脚注、部分以及需要编号的任何其他文档内容。

计数器复位和递增

该`counter-reset`属性适用于任何元素，并将一个或多个计数器初始化为指定值，如果未指定值，则属性为零。

除非重置计数器创建嵌套计数器，否则计数器的范围将包括当前元素及其所有以下兄弟节点。

该`counter-increment`属性适用于任何元素，并且通过指定的值递增一个或多个计数器，如果没有指定值，则属性为1。

该`counter-increment`和`counter-reset`性能上，它的元素忽略display属性取值none。

```
div.example1 { counter-reset: h3 h4; }
div.example1 h3 { counter-increment: h3; }
div.example1 h4 { counter-increment: h4; }
div.example1 h3::before { content: counter(h3); }
div.example1 h4::before {
    content: counter(h4, lower-alpha);
}
```

这将在div元素的范围内创建两个计数器。全部h3和h4标题元素将从1开始编号，计数器编号将放在标题文本之前。

```
div.example2 { counter-reset: h3; }
div.example2 h3 {
    counter-increment: h3;
    counter-reset: h4;
}
div.example2 h4 { counter-increment: h4; }
```

```
div.example2 h3::before { content: counter(h3); }
div.example2 h4::before {
    counter(h4, lower-alpha);
}
```

在此示例中，h4计数器将在每个h3元素上重置 ，以产生在每个新部分重新启动的子部分编号。

嵌套计数器

如果一个元素上的一个计数器被复位，同一个计数器也被该元素的祖先复位，那么将创建一个嵌套的计数器。嵌套计数器的范围仅限于当前元素，其最终值将不会传入其以下兄弟节点。

例如，一个嵌套XHTML列表ul内的元素li内的另一个ul，创建嵌套list-item计数器。

```
ol { counter-reset: list-item; }
li { counter-increment: list-item; }
li::marker {
    content: counters(list-item, ".", decimal);
    color: brown;
    font-weight: bold;
}
```

计数器样式

计数器值默认显示为十进制数，但可以使用其他样式（如罗马数字或字母表的连续字母）显示。

```
chapter { counter-increment: chapter-num; }
chapter::before {
    content: "Chapter " counter(chapter-num, upper-roman);
}
```

该规则将在每章之前生成诸如「Chapter」之类的文本，相应章节号以大写罗马数字显示。

计数器样式见3.4.7一节。

在页面区域中使用生成的内容

页面区域中生成的内容可能包含从文档复制的文本，页码或文本内容。

```
@page {
    @top-left {
        content: "TOP SECRET";
```

```
        color: red;
    }
    @bottom-right {
        content: counter(page);
        font-style: italic;
    }
}
```

@page规则指定左上角的页面区域将包含红色的文本「TOP SECRET」，右下角的区域将以斜体字体包含当前的页码。

从文档复制文本内容

页面区域中生成的内容还可能包含使用string-set属性从文档复制的文本内容：

```
@page {
    @top { content: string(doctitle); }
}
h1 { string-set: doctitle content(); }
```

@page规则指定顶部中心页面区域将从文档中的h1元素中复制的文档标题的内容。

8.7.10 高级分页媒体

命名页

有时需要根据其内容来设计页面。命名页面允许我们选择包含特定元素的页面。我们在重新启动页面编号中使用了这一点，将样式应用于目录页面，以使用罗马数字打印其页码。

包含目录的元素具有page应用的属性，指定该元素的页面是目录页面。该`page`属性可以应用于正常流中的非浮动环境中的任何块级元素。

```
.contents {
    display: block;
    page: table-of-contents;
}
```

然后使用此页面名称将不同的`@page`规则应用于目录页面：

```
@page table-of-contents {
    @top { content: "Table of Contents"; }
    @bottom { content: counter(page, lower-alpha); }
}
```

多个元素可以属于同一个名称，换句话说，页面名称就像CSS类一样。

如果您希望将样式应用于每一章的第一页，则必须使用该`prince-page-group`属性来创建页面组：

```
div.chapter {
    page: chapter;
    page-break-before: right;
    prince-page-group: start;
}
@page chapter {
    @bottom { content: counter(page); }
}
@page chapter:first {
    @bottom { content: normal; }
}
```

该属性`prince-page-group: start;`指示Prince开始一个新的页面组。这是必要的选择，div.chapter:first以在第一章匹配每章的第一页，而不是仅在第一页中。

页面组

当连续的元素属于相同的命名页面，但逻辑上分开的结构（例如单独的章节）将它们组合成一个页面组。这使得它将:first页面选择器应用于整个页面组的第一页（第1章的第一页）。相反，我们通常想要:first应用到每一章的第一页。

这也可能发生：

- 当章节被一个元素分隔，例如h1可能page-break-before: right应用的元素；
- 或者每章是一个块元素，如div。

Prince提供`prince-page-group`可用于启动新页面组的属性。

```
div.body h1 {
    page-break-before: right;
    prince-page-group: start;
    ...
}
```

`prince-page-group`也强制分页，覆盖任何试图避免分页符的属性。left和right页面优先采用`prince-page-group`。

折页和装饰

当一个页面（或列）中断发生在一个框（如一个div）内时，它的底部和顶部边框被克隆，使它们也出现在第一个部分的底部和框的第二个部分的顶部。这是默认值，但可以使用`box-decoration-break`属性设置，将其值设置为clone。

```
div.box1 {
    box-decoration-break: clone;
}
```

如果您希望边框保持打开状态，就好像盒子被简单地切成两半一样，然后可以将此属性设置为slice。

```
div.box2 {
    box-decoration-break: slice;
}
```

竖页打印大画幅的内容

有时需要旋转块元素使其适合页面。这与表格很常见。图表打印一张大表格，显示一张表格，旋转使其宽度适合页面的长度。这可以通过以下规则来实现：

```
@page big_table {
    prince-rotate-body: landscape;
    prince-shrink-to-fit: auto;
}
table.big {
    page: big_table;
}
```

该`prince-rotate-body`属性仅在`@page`规则内工作。然后页面的`@page`规则big_table使用`prince-rotate-body`属性来告诉Prince页面的正文，但不是页眉和页脚，应该旋转。在这个例子中的表格太宽了，所以我们也使用这个`prince-shrink-to-fit`属性使它更小一些。

旋转内容的另一种方法是通过使用`writing-mode`属性更改写入模式，或者通过使用`transform: rotate()`——转换表单元格中的内容来转换元素。

8.7.11 PDF书签

Prince可以创建链接到文档内容的PDF书签。

书签级别

PDF书签具有将它们放置在书签层级中的数字级别。例如，h2的书签可以包含h3或任何更高级别的嵌套书签。使用`prince-bookmark-level`属性控制书签的级别，此处显示此属性应用于XHTML标题元素：

```
h1 { prince-bookmark-level: 1; }
h2 { prince-bookmark-level: 2; }
h3 { prince-bookmark-level: 3; }
h4 { prince-bookmark-level: 4; }
h5 { prince-bookmark-level: 5; }
h6 { prince-bookmark-level: 6; }
```

此属性的默认值为none，禁止为元素创建书签。

`prince-bookmark-state`属性可以控制书签的状态，即书签是否处于open或closed状态。

```
prince-bookmark-state: open | closed
```

书签标签

PDF书签具有文本标签，默认情况下是从生成书签的元素的文本内容中复制的。可以使用`prince-bookmark-label`属性控制此标签的文本，此处显示的属性应用于具有标题属性的章节元素：

```
chapter {
    prince-bookmark-level: 1;
    prince-bookmark-label: attr(title);
}
```

此属性可以包含任何内容值，包括文本字符串和计数器。

书签目标

PDF书签是在激活时显示文档的特定部分的链接。默认情况下，书签将链接到生成书签的元素，这是从标题或章节元素生成的书签的明智行为。还可以使用`prince-bookmark-target`直接使用URL的属性或包含URL的属性来更改书签的目标：

```
bookmark { prince-bookmark-target: url(#intro); }
bookmark { prince-bookmark-target: attr(href); }
```

该属性的默认值是self引用生成书签的元素。

8.7.12 ePub转PDF演示

本节内容根据上面学习的内容，演示如何把一个ePub电子书打印成PDF文件，你可以下载我做的一本电子书，跟着下面的教程，一步步自己动手写代码。

首先在打印PDF之前，我们对整本书进行字体内嵌。教程可见8.4一节。这里提供的电子书是已经子集过字体，所以这一步你可以跳过。

因为我们主要针对的是CJK语言打印，所以我们需要对正文部分，添加齐头尾规则。请把下面的代码添加到正文样式中。

```
prince-text-justify: prince-cjk; /*定义CJK语言的文本对齐方式*/
```

设置封面

封面我们需要做到的效果时全屏展开，独占整个页面。四周不留空白，所以我们可见8.6.4节，进行封面的设置。

首先说下思路：给封面页设置背景而不是插入图片，所以我们需要对封面进行下加工；页面大小为A5（148mm × 210mm），在加5mm的出血与修剪误差。

CSS代码：

```
<style type="text/css">
  @page {
    size: A5; /*A5:148mm × 210mm*/
    margin: 0; /*清除外边距*/
    background: url("../Images/cover.jpg"); /*载入封面图像*/
    background-position: center; /*封面图片水平垂直居中*/
    background-size: 104%; /*放大封面图片使之可以延伸到出血区域*/
    background-repeat: no-repeat; /*封面图片不重复*/
    marks: crop cross; /*添加修剪标志*/
    prince-bleed: 2.5mm; /*出血+修剪，单边2.5mm；即上下5mm、左右5mm*/
    prince-trim: 2.5mm; /*同上*/
  }
</style>
```

HTML代码：

```
<body>
  <p> </p>
```

```
</body>
```

版权页

版权页我们不需要页码，所以我们要为这一页单独声明下计数器。同时不要忘了设置页面大小。

```
<style type="text/css">
  @page {
    size: A5;
    marks: crop cross;
    prince-bleed: 2.5mm;
    prince-trim: 2.5mm;
  }
    @page {
      @bottom { content: normal; } /*清除显示在底部的页码*/
    }
</style>
```

目录页

ePub电子书有两种目录，一种是阅读器可以读取的ncx目录，一种是可以直接被读者看到的HTML目录。当我们需要打印PDF时，就需要先使用Sigil生成一个HTML版本的目录。然后在这个目录的基础上进行格式美化，这里不讲，请参考前文自行设计目录页样式。

```
<style type="text/css">
  .toc {
    display: block;
    page: table-of-contents; /*命名当前网页*/
  }
  @page table-of-contents {
    @top {
      content: "目次"; /*在页面顶部打印「目次」这个文本*/
      font-family: "小标宋"; /*该文本字体是小标宋*/
      font-size: .85em; /*字体大小是0.85em*/
      color: #a52a2a; /*字体颜色*/
    }
    @bottom { content: counter(page ,upper-roman); } /*在页面底部使用大写罗马字符打印页码*/
    size: A5;
    marks: crop cross;
    prince-bleed: 2.5mm;
    prince-trim: 2.5mm;
  }
</style>
```

然后是HTML代码，类.toc添加给body元素：

```
<body class="toc">
...
</body>
```

这我所使用的案例中，我删去了列与列之间的分割线，原因是效果好像并不是特别好。

序言页

一般序言页与正文用的是同一种页码方式，所以我们可以给当前页重新启动页面计数器。

```
<style type="text/css">
.background-img-center {
    display: block;
    page: foreword; /*给当前网页命名*/
    counter-reset: page 1; /*重启计数器*/
}
.forword-title {
    string-set: chapter-title content(); /*标记标题，以共content使用*/
}
@page foreword {
    size: A5;
    marks: crop cross;
    prince-bleed: 2.5mm;
    prince-trim: 2.5mm;
}
@page:left {
    @top-right {
      content: string(chapter-title); /*给标题文本做标记，在左页顶部右侧打印*/
      font-family: "小标宋";
      font-size: .85em;
    }
    @bottom {
      content: counter(page); /*在左页底部打印页码*/
      font-family: "宋体";
      font-size: .85em;
    }
}
@page:right {
    @top-left {
      content: string(chapter-title); /*给标题文本做标记，在右页顶部左侧打印*/
```

```
      font-family: "小标宋";
      font-size: .85em;
    }
    @bottom {
      content: counter(page);
      font-family: "宋体";
      font-size: .85em;
    }
}
</style>
```

相关的HTML代码：

```
<body class="background-img-center" style="padding: 0;">
  <div class="foreword-box">
    <h1 class="forword-title"><span class="foreword-content">序言</span></h1>
    ...
  </div>
</body>
```

篇章页

对于篇章页，一般不会再页面中出现计数的页码。所以我们需要单独给篇章页HTML文件添加下面的代码，清除计数页码：

```
@page {
    @bottom { content: normal;} /*清除页面底部的页码计数器*/
}
```

上面的代码，需要添加到每个篇章页HTML文件中！

正文页面

正文页面仍然采用从序言开始的计数页码，在本书的CSS样式表中添加如下代码：

```
@page {
    @bottom {
      content: counter(page); /*在页面底部显示页码*/
      font-family: "宋体";
      font-size: .85em;
    }
    size: A5;
    margin-top: 16mm; /*天头：书页上端的空白处*/
    margin-bottom: 28.5mm; /*地脚：版心下边沿至成品边沿的区域*/
    margin-outside: 22.5mm; /*订口：书刊需要订联的一边、靠近书籍装
```

```
订处的空白*/
    margin-inside: 17mm; /*切口：裁切书页一边的空白处*/
    marks: crop cross;
    prince-bleed: 2.5mm;
    prince-trim: 2.5mm;
}
```

然后还需要分别在左页右页的页眉部分分别显示书名和章节名，代码如下：

```
@page:right {
    @top-left {
      content: "成语由来"; /*在右页顶部左侧打印书名*/
      font-family: "小标宋";
      font-size: .85em;
    }
}
@page:left {
    @top-right {
      content: string(chapter-title); /*使用二级标题文本，打印在左页顶部右侧*/
      font-family: "楷体";
      font-size: .85em;
    }
}
```

给二级标题添加这个代码：

```
string-set: chapter-title content(); /*给所在样式添加标记，以共content使用*/
```

到了这一步，就完成了ePub电子书到PDF的打印。这是我使用上述代码打印的案例PDF版本，你可以下载下来参考一下。PDF截图：

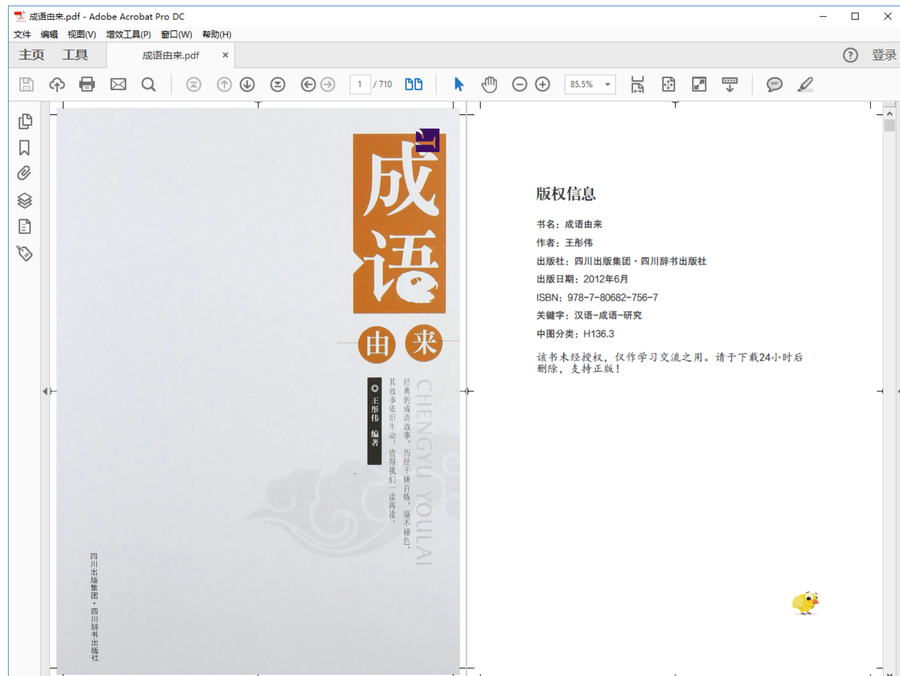


图8-81 封面（Prince水印可在Acrobat中删除）和版权页

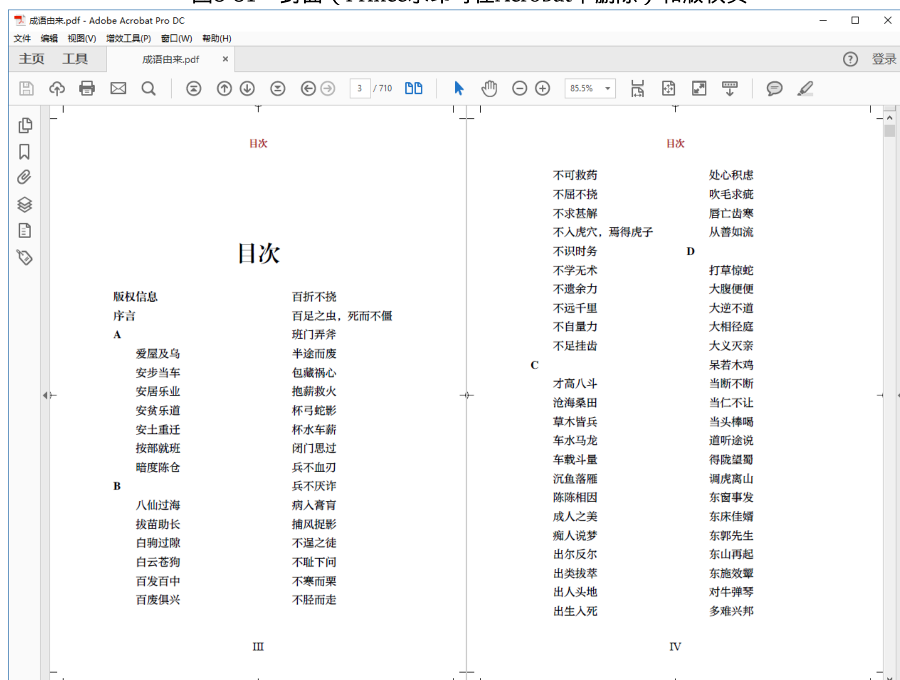
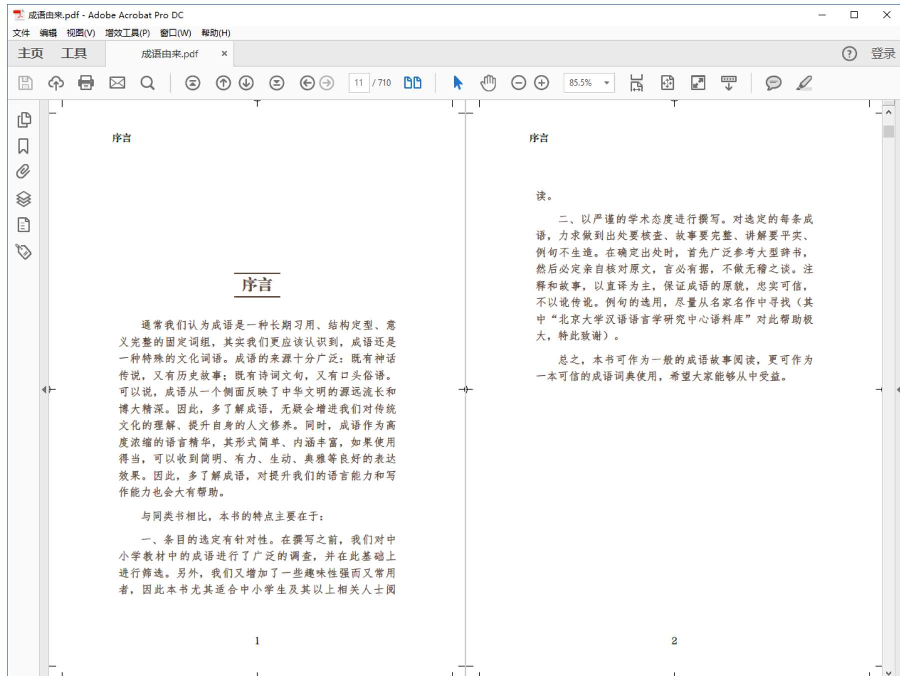


图8-82 目录



序言

序言

通常我们认为成语是一种长期习用、结构定型、意义完整的固定词组，其实我们更应该认识到，成语还是一种特殊的文化词语。成语的来源十分广泛：既有神话传说，又有历史故事；既有诗词文句，又有口头俗语。可以说，成语从一个侧面反映了中华文明的源远流长和博大精深。因此，多了解成语，无疑会增进我们对传统文化的理解、提升自身的人文修养。同时，成语作为高度浓缩的语言精华，其形式简单、内涵丰富，如果使用得当，可以收到简明、有力、生动、典雅等良好的表达效果。因此，多了解成语，对提升我们的语言能力和写作能力也会大有帮助。

与同类书相比，本书的特点主要在于：

一、条目的选定有针对性。在撰写之前，我们对中小学教材中的成语进行了广泛的调查，并在此基础上进行筛选。另外，我们又增加了一些趣味性强而又常用者，因此本书尤其适合中小学生及其以上相关人士阅

1

序言

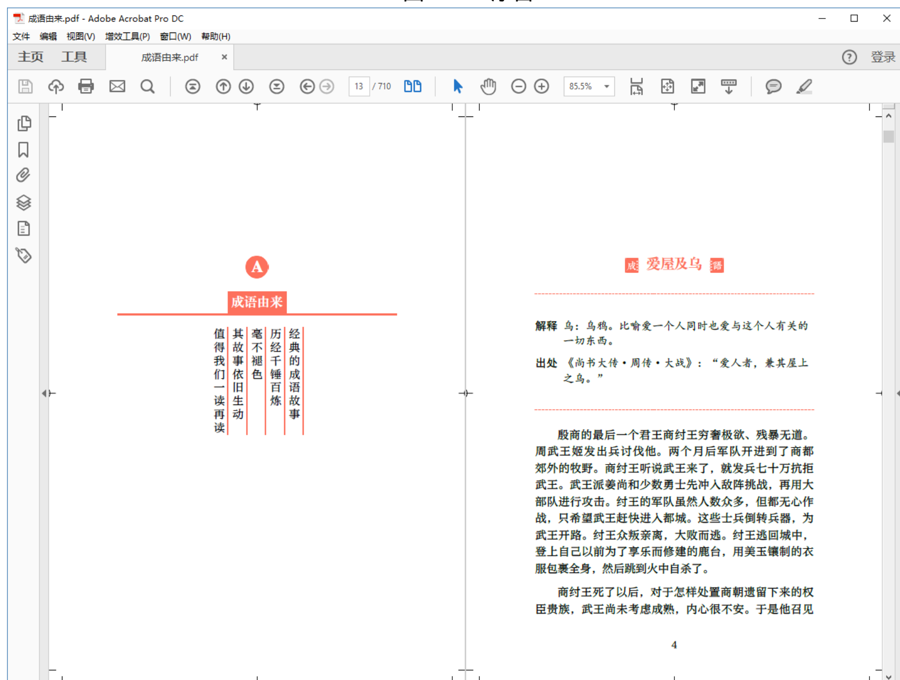
读。

二、以严谨的学术态度进行撰写。对选定的每条成语，力求做到出处要核查、故事要完整、讲解要平实、例句不生造。在确定出处时，首先广泛参考大型辞书，然后必定亲自核对原文，言必有据，不做无稽之谈。注释和故事，以直译为主，保证成语的原貌，忠实可信，不以讹传讹。例句的选用，尽量从名家名作中寻找（其中“北京大学汉语语言学研究中心语料库”对此帮助极大，特此致谢）。

总之，本书可作为一般的成语故事阅读，更可作为一本可信的成语词典使用，希望大家能够从中受益。

2

图8-83　序言



A

成语由来

经典的成语故事
历经千锤百炼
毫不褪色
其故事依旧生动
值得我们一读再读

成 爱屋及乌 语

解释 乌：乌鸦。比喻爱一个人同时也爱与这个人有关的一切东西。

出处 《尚书大传·周传·大战》：“爱人者，兼其屋上之乌。”

殷商的最后一个君王商纣王穷奢极欲、残暴无道。周武王姬发出兵讨伐他。两个月后军队开进到了商都郊外的牧野。商纣王听说武王来了，就发兵七十万抗拒武王。武王派姜尚和少数勇士先冲入敌阵挑战，再用大部队进行攻击。纣王的军队虽然人数众多，但都无心作战，只希望武王赶快进入都城。这些士兵倒转兵器，为武王开路。纣王众叛亲离，大败而逃。纣王逃回城中，登上自己以前为了享乐而修建的鹿台，用美玉镶制的衣服包裹全身，然后跳到火中自杀了。

商纣王死了以后，对于怎样处置商朝遗留下来的权臣贵族，武王尚未考虑成熟，内心很不安。于是他召见

4

图8-84　篇章页和正文

实际上「序言」应该在目录之前，与目录共用一种计数器，然后正文单独一种计数器。

8.8 Web Equation——ePub中插入数学公式

在ePub 2中，出版物可能包含采用图像或SVG格式的数学公式内容。尽管ePub中存在SVG数学公式内容，但对于屏幕阅读器来说，这些内容很难处理，二次编写也非常困难。图像的情况更糟，它们的可访问性（不能复制）更差，而且在不同的字体下和屏幕尺寸间缩放的能力也更差。

ePub 3将MathML作为原生ePub内容类型。可以在不转为另外一种格式的前提下提供MathML标记。MathML语言比较复杂繁琐，学习难度较大且使用频率较低，所以推荐使用专业工具来编写公式，然后输出MathML代码。

Web Equation是一个非常方便的网络服务，可以用鼠标绘制公式，并生成LaTeX和MathML（数学置标语言 Mathematical Markup Language，是一种基于XML的标准，用来在互联网上书写数学符号和公式的置标语言）代码。

推荐使用Google Chrome或Mozilla Firefox打开这个网站。

打开后选择Math图标，进入公式绘制页面。

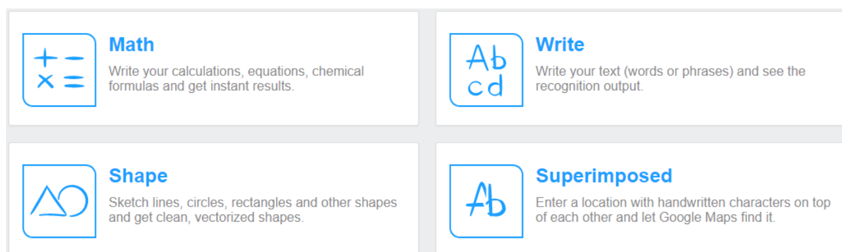


图8-85 Web Equation网站工具

流程：使用鼠标绘制你的公式，完成后会在右上角窗口中显示LaTeX代码，你可以切换到MathML标签，点击右下角的「复制」按钮，复制MathML代码粘贴到电子书中。

绘制公式

使用鼠标依次书写数学符号，并在书写的过程中注意上面显示的公式。发现与实际公式有出入后，点击左上角的「撤销」按钮，重新书写。如果整个公式都识别错误了，可以点击「删除」按钮。

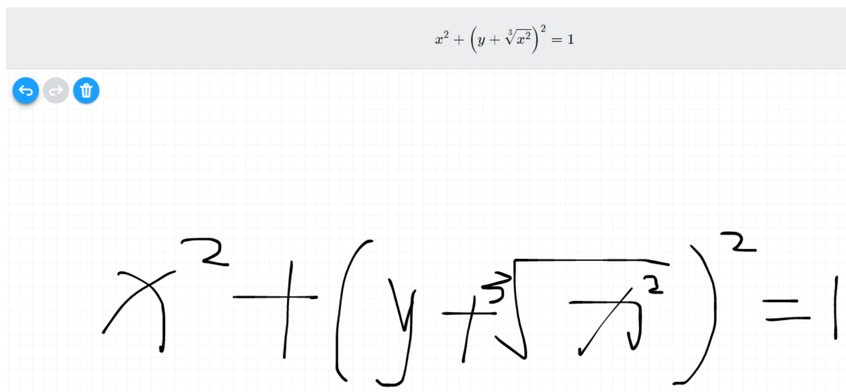


图8-86　鼠标书写公式

复制公式代码

点击右上角的「MathML」标签，然后会在小窗口中显示公式的MathML代码。

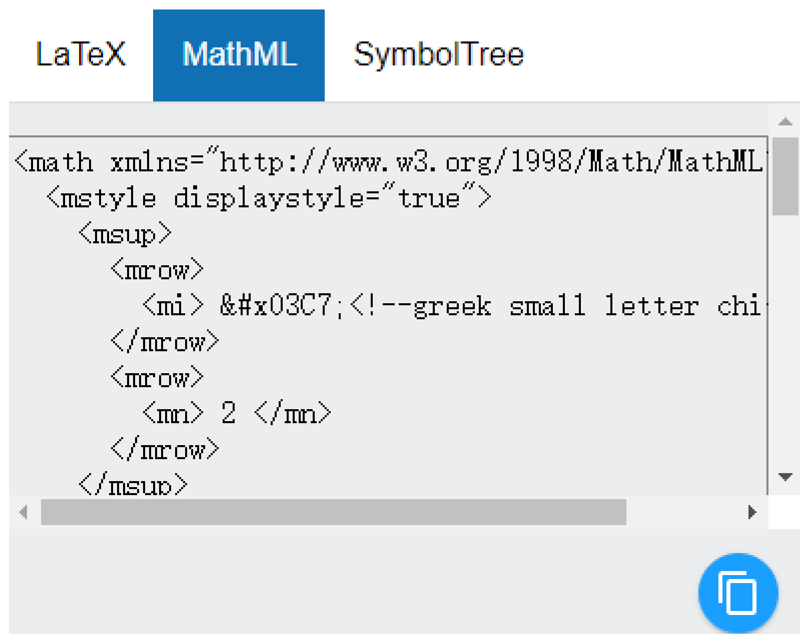


图8-87　MathML公式代码

点击右下角的「复制」按钮，复制代码。在文中需要插入公式处粘贴复制的公式代码。

```
<p>心形线，是一个圆上的固定一点在它绕着与其相切且半径相同的另外一个圆周滚动时所形成的轨迹，因其形状像心形而得名。
  <epub:switch id="mathmlSwitch">
  <epub:case required-namespace="http://www.w3.org/1998/Math/MathML">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mstyle displaystyle="true">
        <msup><mrow><mi>x</mi></mrow>
        <mrow><mn>2</mn></mrow></msup>
        <mo>+</mo>
        <msup><mrow><mfenced><mrow><mi>y</mi>
        <mo>+</mo>
        <mroot><mrow><msup><mrow><mi>x</mi></mrow>
        <mrow><mn>2</mn></mrow></msup></mrow>
        <mrow><mn>3</mn></mrow></mroot></mrow></mfenced></mrow>
        <mrow><mn>2</mn></mrow></msup>
        <mo>=</mo>
        <mn>1</mn>
      </mstyle>
    </math>
  </epub:case>
  </epub:switch>
</p>
```

目前只有比较少的阅读器才可以显示MathML语法，所以你还需要提供一个备用方案来兼容ePub 2阅读器。

```
...
</epub:case>
  <epub:default>
    <span><img class="height-2em" alt="公式图" src="../Images/MathML.png"/></span>
  </epub:default>
</epub:switch>
...
```

即使阅读系统不支持MayhMl语法，也可以显示<epub:default>内的内容。

样式height-2em是公式图所用的样式，一般建议给公式设置高度即可。

尽管MathML是一种核心内容类型，但您必须在OPF文件中声明哪些XHTML页面包含MathML。如下所示：

```
<spine toc="ncx">
  ...
  <itemref idref="MathML.xhtml" properties="mathml"/>
  ...
</spine>
```

某些Web浏览器现在支持HTML 5上下文（无名称空间）中的MathML，但ePub 3要求在名称空间中声明MathML内容。如下：

```
<html xmlns="http://www.w3.org/1999/
xhtml" xmlns:epub="http://www.idpf.org/2007/
ops" xmlns:m="http://www.w3.org/1998/Math/MathML">
```

下载MathML案例电子书，在Gitden Reader中打开，显示如下：

数学公式测试，在lithium阅读器上无法显示；gitden可以。

能把圆周率和e联系起来的初等公式在数学界是少之又少，是数学王国中的国宝级公式。$\pi^4 + \pi^5 \approx e^6$

话说世人皆知勾三股四弦五，而鲜有知道这个简单等式的。$3^3 + 4^3 + 5^3 = 6^3$

这个公式来自于印度数学奇才拉马努金。

$$\sqrt{1 + 2\sqrt{1 + 3\sqrt{1 + 4\sqrt{1 + \ldots}}}} = 3$$

心形线，是一个圆上的固定一点在它绕着与其相切且半径相同的另外一个圆周滚动时所形成的轨迹，因其形状像心形而得名。$x^2 + \left(y + \sqrt[3]{x^2}\right)^2 = 1$

质能方程，$E = mc^2$

图8-88　Gitden阅读器中显示公式

在前文中我推荐了几款阅读器，可以显示数学公式的只有Gitden。

第9章

排版知识

排版，把文字、表格、图形、图片等进行合理的排列调整，使版面达到美观的视觉效果。ePub的排版主要靠HTML和CSS来完成的，这与传统的出版流程不同。传统的排版是固定在一个具体页面中，而ePub的排版则要复杂得多，这是因为用户的显示屏尺寸有不同的大小。所以ePub的最终呈现是流式的、版面不确定的。

下面的文章就试着从传统的排版、Web排版中吸取营养，来更好地服务于ePub。

9.1 Web排版

万维网联盟（World Wide Web Consortium，简称W3C）在2017年2月20日更新了一个文档——中文排版需求。这个需求书总结了一些中文排版的特点，感兴趣的可以看看，这里不再贴出。因为后面会以实际的案例来讲解怎么利用HTML和CSS来排版。

这里分析下Web与数字出版物的差别。

本文章来自由「中国电子技术标准化研究院」「W3C/北航」在2014年9月11日于北京举办的「W3C中文版式标准需求技术研讨会」。作者：北京北大方正电子有限公司 排版软件开发部部长 杨雷鸣（注：有增改）

表13　Web与数字出版的区别、联系

区别	
Web标准	数字出版物标准
主要是网页	主体是电子书
版式、内容简单	版式、内容复杂
内容关联性较为松散	内容的逻辑性、关联性强
版式随意	版式严谨
无明确的（书籍）页概念	传统的页呈现有助于阅读
复杂版面以小块拼大版的方式，正文内容（新闻稿）以简单图文为主，垂直流动	内容按页流动的方式
很少具有附加元素	具有较多的附加元素（目录、脚注、索引、书眉、页码、边文）

联系
行、段排版的规则可以共用
多媒体的支持、脚本的支持可以共用
数学、化学、表格、图表的处理，可以共用技术
电子书的呈现应该以Web标准为基础进行扩展，便于电子书阅读器的开发和技术共享

局部的处理规则可以共享，整体的处理有很大的差异（作为一个出版物有其自身的特点及整体性）。

过去印刷出版物的版式，不要从表面上理解其布局，而是应该分析这种布局给阅读所带来的便利。在数字出版物中，如何实现这种便利。如何充分利用数字出版物尺寸可变、可搜索、可跳转、多媒体等特性来合理

的实现这种便利。要理解印刷出版版式设计的精神而不是照搬其形式。在此基础上利用智能设备和数字出版物的特点，发展新的、有利于用户阅读和知识获取的新的技巧，并将其上升为标准。

9.2 平面设计4原则

排版在我看来就是设计，如果通过排版，让读者更容易的接收书本的内容，这与设计的目的一样。有一本书叫《写给大家看的设计书》（*The Non-Designer's Design Book*）。这是一本实用性远大于理论性的设计书。书如其名，即使是没有受过设计专业教育的人也能从中学到非常实用的设计理念和方法。这本书提出了设计的4个基本原则：

对比（Contrast）

对比的基本思想就是要避免页面上的元素太过相似。如果元素（字体、颜色、大小、线宽、形状、空间等）不相同，那就干脆让它们截然不同。对比能够让讯息更准确的传达，内容更容易的被找到、被记住。如果你想让对比效果更明显，就一定要大胆，不要让两种颜色看起来好像差不错又不一样。当然也不要在同一个页面使用太多种字体。

在排一本书时，可能会遇到明显区分与书籍正文的内容，比如书信、引文等。为了强调这种内容，我们可以改变它的字体、字色、字号、背景色等，或者添加额外的修饰。这就是对比原则的应用。纸书中一般的做法是引文的段落左边界为2汉字字符，首段末段与正文段落扩大间距，字体使用非正文字体。

重复（Repetition）

重复的目的就是「一致性」，让设计中的视觉要素在整个作品中重复出现。可以重复颜色、形状、材质、空间关系、线宽、材质、空间等。这样一来既能增加条理性，又能增加统一性。

电子书作为一个整体，要求我们在排版中要确定整体的风格，即每章、每中类型的段落风格需要统一。它可以是色系一致、正文字体使用同一种字体、段与段间距固定一种、章封设计风格统一等。确定好风格之后，我们可以先排一章，然后调试好效果之后，再配合正则表达式应用到全书中。这是重复原则的应用。

对齐（Alignment）

任何东西都不能在页面上随意安放。每个元素都应当与页面上的另一个元素有某种视觉联系。这样能够建立一种清晰、精巧而清爽的外观，提升可读性。避免一个页面上混用多种对其模式，也就是不要有一些置左，有一些置右。尽量避免使用居中对齐，除非是比较正式、稳重的设计（别动不动就让什么都居中，有些时候居中是一种很土的对齐方式）。

比如格式工整的古诗，一般是五言七言，这种可以考虑使用居中排版。不过，当古诗中有注释图片时，再使用居中对齐，就会存在参差不齐的情况。比如：

床前明月光，疑是地上霜。

举头望明月，低头思故乡。

这是因为注释图片也占了一个字符，所以就出现对不齐的状况。解决办法就是给第2行末尾添加一个全角空格。

床前明月光，疑是地上霜。

举头望明月，低头思故乡。

HTML代码：

```
<div class="JingYe">
   <p>床前明月光，疑是地上霜<img style="width: 1em;" alt="note" src="../Images/note.png"/>。</p>
   <p>举头望明月，低头思故乡。 </p>
</div>
```

` `即是一个全角空格的字符实体。见2.4一节。

亲密性（Proximity）

彼此相关的项应当靠近，归组在一起。如果多个项相互之间存在很近的亲密性，它们就会成为一个视觉单元，而不是多个孤立的元素。这有助于组织信息，减少混乱，为读者提供清晰的结构。

中文的首行缩进2字符，是作为一种区分段落的手段之一。除此之外我们还可以给每段的间距扩大也可以作为一种手段来区分段落。这就是亲密性的体现。另外在写诗歌样式时，需要分段的现代诗，我们也可以根据亲密性这一原则，进而可以确定诗歌首段末段，段与段之间的间距等样式。

Robin（本书作者）所提出的设计的四大基本原则虽然看似简单，却是其他设计理论的基础。比如单就网页的文章排版来说，遵从这四个基本原则，就完全可以做出一个比较好的页面排版效果。然而知易行难，只有在平时的生活、工作和学习中多留心、多思考、多实践、多总结，才能真正领会本书的真正用心，提高一点审美品味和设计的基本能力。

如果说文章是作者和读者之间的一座桥梁，那么糟糕的排版设计就是一条独木桥，而好的排版设计是宽阔坚实的现代化大桥。我觉得，所谓好的阅读体验无外乎两点，一是内容的易读性，一是信息尽量准确无误的传达。而好的排版设计，也应该实现这两点。

9.3 术语

虽然传统出版与电子书的编辑因为载体的原因有很大的不同，不过我们仍然可以从传统出版中获取关于排版知识的营养。

封一。封一是封面或封皮的术语。封面分为骑马订、平装（订）、假精装和精装等装订形式。封面内容包括书籍的种类、书名、著译者名、出版社名以及标识书籍内容的图标等。

封二。封二是封里的术语。封里分为印有内容和没有内容两种形式，期刊杂志一般印有内容，书籍一般没有内容。

封三。封三是封底里的术语。印刷的内容和版心同封二。

封四。封四是封底的术语。封底右下角印有书名、定价、条码、上架建议等；封底左上角印有责任编辑、责任校对、封面设计等。

书脊。封一和封四订口一侧的联结处，也称为书背（后背），精装书有圆背和方背两种形式。书脊宽在5mm以上的印有内容，如丛书编号、书名、著译者名、出版社等。

勒口。平装封面的一种形式，封一和封四切口一侧向内折过的部分，宽度在5cm左右。分为印有内容和没有内容两种形式，一般会有作者简介、内容提要、系列书书目等。

扉页。扉页又称为内封、副封面，扉页内容与封面相同，但是扉页内容排版的字号要小于封面，扉页的位置在封面后正文前。

衬页、环衬。封面和扉页、封底和书芯之间的空白页称为「衬页」（纸的品种可以不同），一张的称为衬页，两张的称为「连环衬页」环衬。

隔页。隔页又称为篇章页、中扉页，一般占两个暗码。如果隔页印刷的墨色或用纸与正文不同，隔页按无码处理。隔页是书籍标题（一级）排版位置的一种形式，因为隔页上的内容一般是篇、章的标题，所以也称为「篇章页」；它又是书名下的第一层标题，所以也称为「中扉页」；它又对书籍阶段性内容起到隔离作用，所以称为「隔页」。

版面。版面是书刊一面的幅面，成品开本尺寸的大小。版面中间包括版心中的文字、表格和插图等内容及版心四周的白边，是计量书籍内容数量的单位。

版心。版面中间的文字、插图、表格等内容部分称为「版心」。版心是书籍内容的排版位置，没有特殊情况内容不能超出版心排版。

设置版心会让版面更具条理性，同时也保证了内容的识别度。所有元素都应该在版心内编排，但并不是绝对不能超出版心，在已有规则下，我们可以将部分重要性低的元素放置到版心之外，从而为画面带来动感与张力。

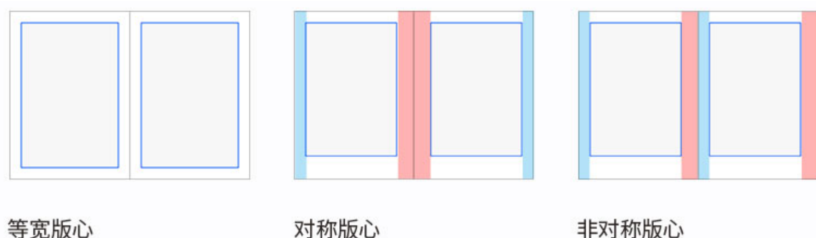


图9-1　版心的分类

等宽版心：四周天头地脚与内外页边距均为相等，版心位于画面的居中位置，版心的大小可以根据项目气质来决定，但对于印刷品来说，与页面边距最小要预留5～10mm。

对称版心：对称版心通常是以订口为轴心水平镜像的对页，改变上下左右的边距，会让版面更具动感与变化性。对于多页的设计，内页边距应该设置的较宽一些，以免装订时版心陷入订口中。

非对称版心：这是非对称版心的设置，大多数情况下，非对称版心都会让对开页的天头地脚保持统一，并在此基础上改变切口与订口边距的比例关系，使其产生不等宽效果。

版口。版心的四周称为「版口」，分为上版口、下版口、右版口和左版口，左版口为默认版口。

订口、切口。版心的里口称为「订口」，版心的外口称为「切口」。也可以说是订联的一边称为「订口」，刀裁的一边称为「切口」。

下面的版面结构，参考8.6.5节可以加深认识。

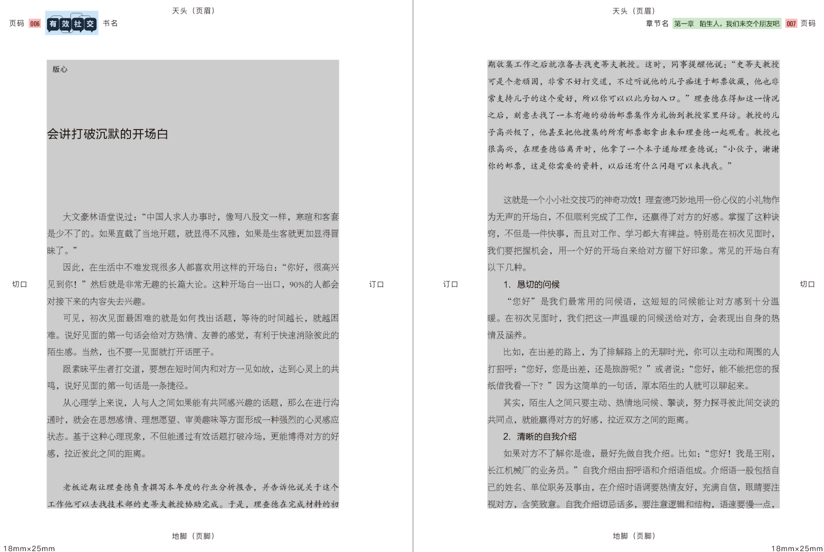


图9-2　图书页面各区域名称

天头、地脚。以版心而言，上版口的位置是「天头」，下版口的位置是「地脚」。以版面而言，版心的上版口距版面成品的边称为「天头」；版心下版口距版面成品的边称为「地脚」。在电子书阅读器中可以通过设置，在顶部显示章节名，在底部显示页数等。

书眉是指版心上端、下端或左右端排的书刊名、篇章名、标题或词条等，用比正文小的字级排列的文字。这些文字可以是书名、篇名、章名或刊名，目的是便于读者查阅，也是版式的一种装饰。一般书心偶数页码版面排书名，奇数页码版面排章名；或者偶数页码版面排篇名（章名、刊名），奇数页码版面排章名（节名、文章名）。即偶数页码版面上排上一级标题做书眉，则奇数页码版面上排下一级标题做书眉。遇有另页或另面时，书眉可省略不排。

书眉所用字号比正文字号小，字体可以变化。书眉下方或书眉后居中用一条线与正文隔开。该线称之「书眉线」，其长度可与版心宽度相同，也可小于版心宽度。书眉文字过长的，可作适当删节，但全书必须一致。

竖排书刊的书眉，排印在切口，线装本书籍书眉排在折缝处中间，所以也称「中缝」。目前一般的书籍不排书眉，只有辞书、工具书、巨著排书眉。为了美观，实践中，书眉设计常有变化和创新。

期刊的书眉与图书书眉不同，它有如下特点：

1. 因期刊无章节而有固定栏目，故书眉多是栏目的名称，也有的偶数页码为栏目名称，奇

数页码为文章名称。

2. 不同栏目可用不同的书眉格式，因此一本期刊中常有多种书眉格式。
3. 书眉文字多加装饰，其位置既可居中，也可靠翻口。
4. 书眉文字多与书眉线排在一行内，也可以排在书眉线之上或之下。
5. 各个栏目对应的书眉格式一经设定，便在较长一段时间内保持相对稳定，在各期期刊中都基本相同。

插页。排版内容超出版面的印页。插页上的内容如果是表格称为插页表，如果是插图称为插页图。插页不排页码（属于无码）。排版内容尽量不排插页，采取其他版式处理方法，如做对照页。因为如果做插页，印刷、装订都需要另外加工。电子书没有这方面的限制。

无码、暗码。本面的内容不排页码也不占页面位置的称为「无码」。例如：排版内容前的页码是「88」，排版内容后的页码是「89」。本面内容不排页码，但占页面位置的称为「暗码」。例如：排版内容前的页码是「88」，排版内容后的页码是「90」。

页码是标明版面顺序的序号。页码一般排在地脚，靠近翻口处，也有排在天头靠近翻口处。若开本较大，也可排在地脚或天头的居中位置。有书眉时，页码与书眉可排在同一行。

页码一般分正文前的页码和正文页码。编排页码时往往从正文的第一页开始到书末最后一页用阿拉伯数字（两边可以加上装饰点或装饰线）连续编号，称之正文页码。遇到篇、章是另页排的，整面的超版口图或表，或章末的空白页则不排页码，而用暗码，即页码连续计数，但不印出，空白页所占的暗码也叫空码。页码用的字号、字体应根据书籍的性质、开本和版式确定。

将正文前的辅文，如主书名页，版权页，附书名页，出版前言，序，目录等（俗称文前部分）用区别于正文页码的数字型式（如罗马字、斜体数字等）连续编号，称之文前页码。

期刊页码的字体、字级大小无一定的限制，可以小于正文、等于正文或大于正文，也可用白体、黑体、正体或斜体各种字体。期刊中用页码加以装饰的情况亦不少见。总的原则是以美观得体为宜。

行。正文加上正文下标注的行距称为「行」。横排书的版心高度（竖版心）、竖排书的版心高度（横版心）以「行」为计量单位，计算机软件称谓「行高」。例如横排版四行：如果行距宽度是字号的1/2，等于6个字高；如果行距宽度是字号的1/4，等于5个子高。

倍。计量字数的单位，一个字就是一倍。一倍也称为「全身」或「全角」，一般「线」的长度或「边」的长度用倍表示。倍这个概念类似CSS单位em。

行距。行在字典里的意思是用于成行的东西，如苏轼的《江城子》：「相顾无言，唯有泪千行。」，上行正文与下行正文之间的距离称为「行距」。



图9-3　行间距

字号24px、行间距36px，测量后两条青色参考线之间的距离是12px。在CSS中行间距用line-height属性控制，它的行间距计算方法和Ps一样，行间空白距离＝行间距-字符高度。电子书排版中的行间距，从1.4em到1.7em均可，以实际的体验为主。

段间距。段落在词典中解释是根据内容划分成的部分，和行间距一样。在电子书中用margin属性控制段落间距离，margin的上下值相同，上段的margin-bottom和下段的margin-top会重合在一起。两段margin值不同，上段的margin-bottom和下段的margin-top会相加。

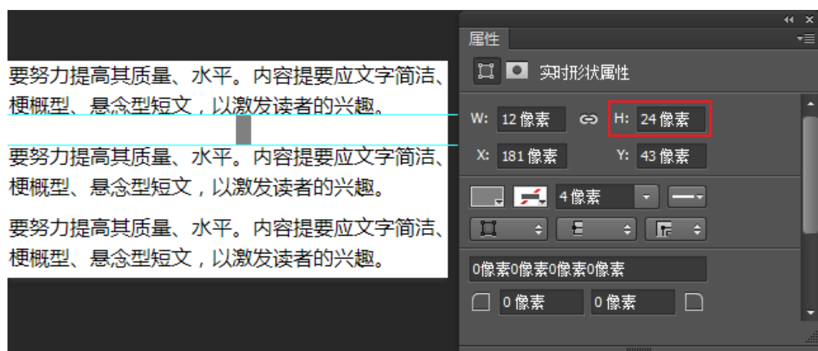


图9-4　段间距

CSS的代码：

```
p {
    line-height: 1.5em;
    margin: .5em 0;
}
```

两端对齐。又称作齐头尾，可参考text-align属性的介绍，两端对齐还要注意避头尾。避头尾最直观的是指一行不以标点做开始，这在我们上学写作文时语文老师都讲过。下面这段文字没有设置两端对齐、避头尾：

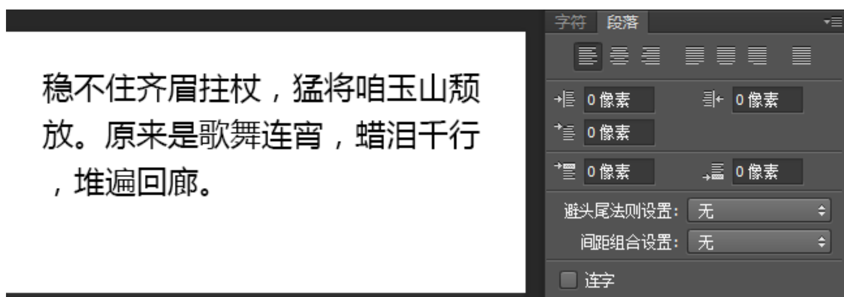


图9-5　段落未避头尾

可以看到第3行是以逗号最为开始的，这种就不合适。然后设置「最后一行左对齐」、避头尾法则设置「JIS 宽松」：

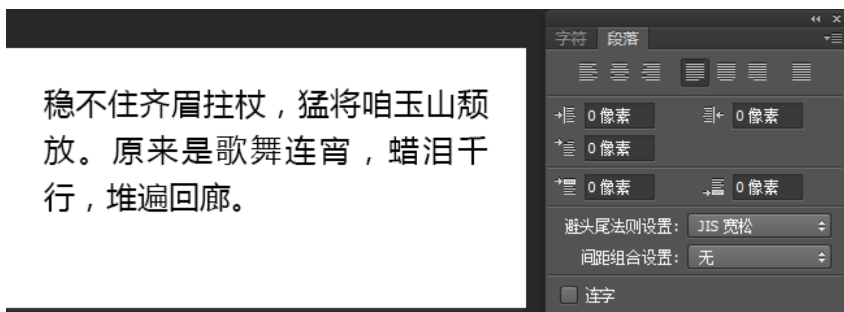


图9-6　段落避头尾

JIS是日本工业标准的简称。通过添加字符间距的方式拉伸段落，把本应该在第2行的「行」字移动到第3行开始。这样看起来就比较舒服了！

基线。计算机排版的定义，即大字号和小字号在一行排版，以底部对齐的，底部的一条直线称为「基线」；大字号和小字号在中部对齐的，中部对齐的假想线称为「中线」。

字距。文字与文字之间的距离称为「字距」。正文正常排版文字与文字之间没有字距，疏排或者标题文字与文字之间就需要有不同的字距。

版式。版式就是排版的格式（规格、标准、规范）。排版时版式是否符

合工艺标准直接影响读者的阅读体验。版式有两层含义。

对整篇（页）而言，版式是排版的位置或者排版的样式。

对局部而言，版式是排版的格式，如标题、正文、表格、数学式、化学式等格式。

另外，版式根据书籍内容的特点还区分为三大类别。

出版印刷物：文艺理论、科技、医药、工具书、外文书、古籍书等；

新闻印刷物：包括大报四开（正统性质）、小报八开（杂志性质）；

社会印刷物：各种证券、说明说、各种零件表格。

出版印刷物和新闻印刷物属于示性印刷物，供读者阅读；社会印刷物供读者使用。排版时还需要注意这一特点。

齐肩排、齐口排。这两种排法是对排版内容回行后位置的要求。

齐肩排：又称为「齐字排」，指自然段的第二行及以下各行，回行与上行除去序号的「字」对齐排。

齐口排：又称为「顶格排」，回行「齐版口」。

1. 齐肩排又称齐字排，指自然段第二行回行后与上行序号后的字对齐。
2. 齐口排又称顶格排，指排版内容第二行回行对齐版口，是最基本的回行方式。

上面的列表就是「齐肩排」，「齐口排」则和正文一样。

行序、字序。指正文「行」和正文「字」的排列顺序。

行序：横排由上至下；竖排由右至左。

字序：横排由左至右；竖排由上至下。

目前，横排书和竖排书的字序和行序，计算机排版软件能自动生成。但是有的排版内容还需要排版操作人员调整顺序，如插图、表格等内容。

横排	竖排				
图1	图2	图2	图1	图3	图1
图3	图4	图4	图3	图4	图2

横排遵循从左到右，竖排遵循从右到左或从上到下的规则。

左翻本、右翻本。是横排书和竖排书的两种翻法。

左翻本：又称为西式翻身，横排书的翻法。书籍的订口在左边，书籍的切口在右边。

右翻本：又称为中式翻身，竖排书的翻法。书籍的订口在右边，书籍的

切口在左边。

大横开本、小横开本。是横排书的两种开本形式，用于不同性质的排版内容。通常的图书一般都是采用小横开本，大横开本则适用于统计表格、机械制图等需要横版心宽的图书。在电子书排版中，则不需要考虑这种情况，比如一张图设置了交互图模式，就能随意地放大缩小，不受屏幕的物理尺寸限制，这也是电子书的优点。

大横开本：横版心长，竖版心短。

小横开本：横版心短，竖版心长。

页、面。计量图书的单位，一页称为两面，两面称谓一页。页是指一张纸，一张纸有前后两面，用P表示，例如24P，印在纸上就是12页（张纸）。

对照页。对照页是处理图书内容超出本心的一种方法，两个版面看成一个版面合排成一个内容，如表格或插图。对照页又称为合页、双跨单、和合页、蝴蝶页。这种排版方式类似多看阅读中的speed效果。

破栏。在双栏以上版面上的插图或表格，本栏排不下，需要占到另一栏的位置称为「破栏」（跨栏）。由于手机屏幕本身就不够宽，所以很少有双栏排版，另外双栏（或称多列属性）也需要阅读器支持。

串文。串文又称为「盘文」，为使版面紧凑，在插图旁或表格旁串排文字。

印张。印张是计量书籍印刷篇幅的单位。以书籍内容印满大纸的一面为一个印张。如32开本32个页码为一个印张，64个页码为两个印张。16开本16个页码为一个印张，32个页码为两个印张。两个印张用一张全开纸。

版权。版权页是出版者的权利和作者的权利页，同时也是图书出版的记录页。包括：开本、印张、字数、版次、印次、印数、定价以及标准书号等内容。

接排、另行、回行。

接排是指原稿另段的情况下与上行连接排版。

另行是指在原稿接排的情况下另行排，也称「另段排」。

回行是指另行的第二行对排版格式的指定要求。

上空、下空，前（左）空、后（右）空。

上空、下空是指排版内容距上或距下的距离。一般以「行」为单位。

前（左）空、后（右）空是指排版内容距左版口或右版口排版的距离。一般以「倍（em）」为单位。

补白。全书的一级标题要求另页或另面排版的，但正文又不允许转页排，正文最后一面如果余有空白，为使版面美观，按空白位置补充适当独立的小段文章称为「补白」。只有在设计固定版式的情况下才需要，而常见的电子书属于流排，即排版内容会根据屏幕的大小自动适应版面。

修订本。修订本指对原书的内容有重大修改，又重印的版本称为「修订本」。

版本。同一部书因为编辑、传抄、刻版、排版、印刷或装订形式的不同而产生的不同的本子，称为「版本」。形成各种版本的因素很多，如成书时间，书的版次，正文的字体，内容的增、删、修改等。

洁本。需要出版有参考价值的书刊，其内容有淫秽或荒诞之处，经审读删节后重印出版的本子。

节本。图书篇幅较大，为适应某一层次读者需要加以删节后印刷的版本。

孤本。独一无二的图书。某书的某一刻本，只流传于世一份的叫「孤本」。又如未刻印的手稿和碑帖的旧拓本，仅有一份的，也称「孤本」。

善本。在几种版本中，经过精加校勘，错讹较少的，称为「善本」。

百衲本。百衲，原指用多块布拼缀成的僧衣，这里指用不同版本的残卷、零页配合后会印而成的一部完整的书籍。

珍本。指珍贵的书籍、资料。凡罕见的、难得的，以及有历史意义的科学、艺术价值的古旧图书资料与革命文献等都属珍本。

凡例。作者向读者介绍图书内容编写的个例规范，以便读者更便捷地学习、参考图书的内容。

体例。对图书版式的总体要求和个例规范。

9.4　字体

汉字是世界上最古老的文字之一，而且每个文字都含有其音形义，从河南安阳殷墟的发掘中可以看出，汉字是三千多年前殷商的「甲骨文」和稍后的「金文」演变而来的。总之，汉字是逐渐由图形变为笔画，象形变为象征，复杂变为简单。

衬线与非衬线字体

衬线体指的是有衬线的字体，又称为有衬线体、衬线字、曲线描边字，俗称白体字；而与之相对的，没有衬线的称为无衬线体、无描边字，俗称黑体字。衬线指的是字型笔画末端的装饰细节部分。无衬线字体在西文中习惯称sans-serif，其中sans为法语的「无」的意思；而另外一些人习惯把无衬线体称grotesque（德语作grotesk）或「哥特体」，把衬线体称为「罗马体」，但是这些词已经不是很常用了，只保留于字体名称中。

中文将serif按照字义翻译成「衬线」或「字脚」，顾名思义就是装饰陪衬的作用，因此有衬线体也称为「有脚体」。日语中将衬线称为うろこ，字面意思为「鱼鳞」。

衬线体（即「白体」），中国的印刷界称之为宋体，港台的繁体中文电脑系统称之为明体，日文称明朝体（みんちょうたい，Minchōtai），韩国语称바탕（Batang）体或명조체（Myeongjoche，同样是明朝体），常用于正文排版。这些名称起源于中国历史上的宋朝和明朝，当时中国的雕版印刷术已经广泛传播，而用于制造活字的木纹多为水平方向，因此造成在刻字时横画细，竖画粗；而且为了防止边缘破损，横画在两端也被加粗，根据运笔习惯而形成三角形的装饰。这种方式一直沿用至今，现在白体类字体多数都是横细竖粗。

另外，无衬线体在中文通常称为黑体，在日文称为ゴシック体（Goshikku-tai，即「哥特体」），韩国语中称돋움（Dotum）体或고딕（Gotik）体，这类字体笔画粗细基本一致，没有衬线装饰，较为醒目，

常用于标题、导语、标志等。

好多近乎知

思源宋体（serif）

好多近乎知

思源黑体（sans-serif）

图9-7　衬线体与非衬线体

在传统印刷中，衬线字体用于正文印刷，因为它被认为比无衬线体更易于阅读，是比较正统的。相对的，无衬线体用于短篇和标题等，能够引起读者注意，或者提供一种轻松的气氛。

一般来说，人们倾向在长篇文章中使用衬线字体，如书籍、报纸和杂志等等。虽然在欧洲比北美更经常使用无衬线体，但在正式场合衬线字体还是使用最多的类型。印刷制品更多趋向使用衬线字体以方便阅读，但是在计算机领域中倾向使用无衬线字体以方便在显示器上显示。

字体嵌入性

在Windows系统中选中一个字体文件，点击鼠标右键，选择「属性」一行。在弹出窗口中的「详细信息」一栏，有一个「字体嵌入性」属性。有4种属性：

- 只读（显示和打印） Only printing and previewing of the document is allowed (read-only)
- 读写（可编辑） Editing of the document is allowed
- 无限制（可安装） Everthing is allowed (installable mode)
- 限制（禁止嵌入） Embedding of the font is not allowed

古今字体

汉字的变化分为古文和金文两大类。古文广义指甲骨文、金文、籀文

（相传是周宣王时太史籀所造，即大篆）以及秦始皇统一中国后统一的文字小篆。金文广义指从隶书到现在通行的文字。楷书就是从隶书发展演变而成的。



图9-8　汉字「车」字形演变（出处）

甲骨文：公元前十四到十一世纪商代统治者的占卜记录，由于刻在龟甲和兽骨上而得名，是中国现存所发现的最古老的汉字。

金文：商代后期流行在青铜器如食具和兵器上铸的铭文，因为先秦称铜为金，所以叫「金文」。战国晚期渐少使用青铜器，金文亦随之衰落。

大篆：通行于春秋、战国时代，其中以《石鼓文》上的石刻大篆为代表，到秦朝仍然通用。

小篆：秦始皇统一中国后，以此统一全国文字，这是中国史上第一次大规模的文字规范工作。汉代篆书是较为庄重的文字，《说文解字》收录了九千三百多个字。直至三国魏晋时期篆书才不再流行。

隶书：秦代官员为方便文书，发展出比小篆简易的隶书，汉朝成为官方文字。因为汉朝立碑的风气盛行，不少隶书石碑流传至今。

草书：从写得很草率的大篆和隶书的基础上发展出来，使用者主要是经常要起草文书的史官。汉元帝时的史游，就用草书体写了《急就篇》。

行书：后汉时期民间流行的新兴字体，以晋代王羲之的《兰亭序》为代表作。

楷书：兴于汉末，普及于唐代。人们嫌隶书太麻烦，而草书又难以辨认，便在隶书的基础上创造出楷书，代表的书法家有欧阳询、诸遂良、颜真卿、柳公权等。

魏碑：南北朝崇尚佛教，建寺造塔，雕刻佛像，并雕刻了许多造像记、

石碑、墓志等。魏碑体基本是隶书向楷书演变而成的，方正严谨。

印刷字体

印刷用字体既要便于阅读，又要符合规范，笔画粗细统一，而且要有艺术性。所以人们在长期的实践中，根据碑体和帖体创造了许多种印刷字体。这些在印刷物上使用的字体，可以统称为印刷字体。现在通用的印刷字体有宋体、黑体、仿宋体和楷体四种基本字体。

宋体

宋体是明朝隆庆、万历年间，写字人按照宋精本的字体雕刻的，故称为宋体，习称老宋，在日本称为明体；因为应用广泛，又称普通体；宋体对黑体而言，又称白体。宋体字形的特征是横细竖粗，字面大，字体端正。它特别适合于排印书刊正文。

「书宋体」距今已有近60年历史了，凭着清晰爽目、久读不易疲劳、印刷适性好的优点，书宋体一直被视作书刊正文的首选字体。方正书宋体，字体端正清秀，横轻竖重，结构均匀，笔法严谨，美观实用，适用于各类书刊、杂志的正文字，也是屏幕用字的上乘之选。（方正书宋）

黑体

日本昔日对各种洋货商标上的拉丁文字造型很感兴趣，于是借鉴产生汉字黑体。三十年代，日本黑体传入中国，称谓中国四大印刷字体之一。黑体又称方体、平体、粗体、方头字，它的名称是根据它的形状而言的。黑体字形的特征是横竖都粗，笔画粗壮，特别醒目。它适合于用做标题或重点语。

汉字的黑体是在现代印刷术传入东方后依据西文无衬线体中的黑体所创造的，黑体字没有衬线装饰，字型端庄，笔画横平竖直，笔迹全部一样粗细。方正黑体笔划均匀，字型端正古朴，构体平稳充实，尤其适用于各类标题字。（方正黑体）

仿宋体

是模仿宋精本雕刻的字体，所以称为仿宋，习称仿体，又称方仿。仿宋字体字形的特征是横竖都细，字形介于宋体、楷体之间，字级清秀美丽。常用做标题下的引文，杂志中也有用这种字体排整段文章的，古书则以用仿宋体的居多，文件也都是用仿宋字体。

现在的仿宋体是1955年以后上海华丰字模的字体。以前的仿宋体只是在宋体字的基础上，把竖笔画改细，字体不太美观。

仿宋体是20世纪初才出现的一种新字体，它是以宋版书中的瘦细体为基

础而设计的，其特色是横竖笔画等粗，整体笔画略细，起笔处有斜势棱角，弯部有棱角分明的字肩，字型挺拔清秀，给人以悦目之感。方正仿宋体笔划粗细均匀，字型俊秀挺拔，刚劲有力，布局严谨，错落有致，清晰美观，容易辨认。（方正仿宋）

楷体

楷体的字形特征是与手写的正楷字基本一致，所以又称手写体、活体、正体。楷体的字体匀整、端正、挺秀悦目，由于和手写体基本一致，所以适合于印制小学课本、少年读物、通俗读物等。但如果用于图书正文，从全版面看，则不如宋体字整齐。

楷书，又称正楷、楷体、正书。「形体方正，笔画平直、可作楷模。故名楷书。」是《辞海》对楷书的解释，它始于汉末，通行至今，长盛不衰。方正楷体字体朴实端正，笔法舒展有力，流畅自然，结构匀称，清秀，平和，书卷味十足。（方正楷体）

以上4种方正字体，在方正字库网站注册后，都可以免费获取个人非商业授权版本。

隶书体

字体的字形与楷体近似，但比楷体字浑厚饱满，笔法古朴典雅，常用于书、报刊的标题。与宋体、黑体、楷体并列称为四种「常用字体」。

牟体

是牟紫东专门为《人民日报》设计的标题字体，有些像拉长的字体（字幅的宽是字号的3/4～2/3）。

报宋、书宋、宋三、小标宋、大标宋

根据计算机排版软件字模压缩的特点和排版的需要，计算机排版软件把宋体又分为：报宋体、书宋体、宋三体、小标宋体和大标宋体。

活字排版的宋体是根据字号大小设计的字模，笔画粗细符合字号幅面的要求。而计算机宋体是根据活字的宋体制作的，报宋和书宋在排大字号时笔画相对较细，所以不同的字号在排版时需要用不同的宋体。

一般排报刊用报宋，排书刊用书宋，四号以下标题可以用书宋，三号标题用宋三，二号标题宋宋三或小标宋，一号以上用小标宋或大标宋。

大标宋，顾名思义，是适合做大标题的宋体。方正大标宋源自1970年上海字模一厂设计的行头宋，是专门为大字号标题设计的，可以说是方正小标宋的姊妹。相对于方正小标宋，方正大标宋的笔画略粗，中宫略为收紧，这符合大字宜紧密，小字宜宽绰的书法规律。但总体而言，这两

款字体风格很接近。方正大标宋字体端庄严谨，笔划粗细分明，布局凝重沉稳，适用于书、报、杂志等的各类标题字。

小标宋，顾名思义，是适合做小标题的宋体。方正小标宋源自1964年上海印刷技术研究所设计的标题宋，是专门针对小字号标题设计的，长期被广泛用于「红头文件」的标题。方正小标宋字型端正，结构匀称，笔划横细竖粗，布局严谨、稳重，适用于书、报、杂志等的小标题字及说明用字。

以上两款方正标宋字体需要购买授权方可下载。

思源宋体和思源黑体

思源宋体（Source Han Serif，Google公司则称Noto Serif CJK，「思源」一词来自于成语「饮水思源」）是Adobe与Google所领导开发的开放源代码字体家族，使用SIL开放源代码字体授权在GitHub上发布。其于2017年4月3日发布，是对应于思源黑体的宋体（明体）字体，支持四种不同的东亚语言（简体中文、繁体中文、日文和韩文），且各有7级字重，其中的每个字重各含65,535个字型。

Adobe与Google发布思源黑体及思源宋体是为了呼应统一设计的需求，以服务于东亚15亿人口，并依照当地语系国家的标准规范，如日文采用日本JIS X 0208及JIS 2004、韩文采用KS X 1001及KS X 1002标准、繁体中文支持Big5并根据中华民国教育部国字标准字体造字、简体中文以中华人民共和国的GB 18030及通用规范汉字表规范制作。

思源宋体的开发由Adobe的西冢凉子主理，拉丁语、希腊语、西里尔文及数字采自Source Serif Pro字体，东亚文字（汉字、假名、谚文）部分则分别以各语种为基础，由不同设计公司制作——中国常州华文、日本Iwata Corporation、韩国Sandoll Communications。

字体的名称，除了英文名Source Han Serif，日文名是「源ノ明朝」、简体中文为「思源宋体」、繁体中文为「思源宋體」、韩文为「本明朝（본명조）」。

思源黑体（Source Han Sans，「思源」一词来自于成语「饮水思源」）是Adobe与Google所领导开发的开源字体家族，1.001及更早版本以Apache 2.0许可证授权，而1.002及更新版本则使用SIL开源字体授权，属于无衬线黑体。思源黑体于2014年7月16日首次发布，支持繁体中文、简体中文、日文及韩文，并且各有7种字体粗细。公开之时为当时涵盖字符数量最多的字体，44,666个字符分属于65,535个字形中，此为OpenType字体技术的极限。

思源黑体的开发以Adobe的西冢凉子为中心，西文及数字采自Source

Sans Pro字体，汉字部分则分别以各语种为基础，由不同设计公司制作——日文为Iwata、中文为常州华文、韩文为Sandoll等，其字形皆依循各国国家标准。共包含7种字重（ExtraLight 100、Light 200、Normal 300、Regular 400、Medium 500、Bold 700、Heavy 900）。

字体的名称，除了英文名Source Han Sans，日文名是「源ノ角ゴシック」、简体中文为「思源黑体」、繁体中文为「思源黑體」、韩文为「본고딕」。

以上两款字体完全免费。字重为Regular。

汉字字体的识别

汉字的字体基本上是由四种基本字体派生的字体。如准圆、大黑、宋黑都是由黑体派生的。彩云和琥珀是在黑体的基础上加上了象形，所以掌握了宋体的特征，字体是不难识别的。

字体在图书中的使用

正文字体的使用，在每个字体介绍中已经讲到。标题用字，见10.2节。



图9-9　方正字库：汉字字体示意图

外文字体

中文字体一般以碑体或帖体命名。而外文字体以字体形状命名。分为白正体、白斜体、黑正体、黑斜体。白正体又称罗马体，白斜体又称意大利体。

不同的外文文种字体的命名也不相同。如德文的花体（哥特体）、拉丁

文的手写体。见外文字母。

9.5　字号

字号是指定印刷字大小的单位。为适应不同图书的性质和图书内容的要求，在图书排版时设计选用不同的字号。我国最早的字号有七种：一号（显字）、二号（明字）、三号（中字）、四号（行字）、五号（解字）、六号（注字）、七号（珍字）。计算机排版开始有22种字号，现已经发展为多种任意字号。

号、磅、像素、em

号。号数制是我国用来计量汉字大小的标准制度，以「号」为单位。号只能表示字大小的级别，但不能同时表示度量标准。

磅。也称为「点」数制（Point，标、博等），是欧美各国用来计量拉丁字母的标准制度（简写Pt或p.），1英寸等于72p，1磅等于0.3527mm，计算机排版软件定为1p等于0.35mm。磅可以直接反映字度量的标准。

对一般成年人而言，10.5磅是常用的标准大小，这个字级在铅字活版时代称为「老五号」字。给儿童和老年人看的书，通常应该用更大的字级。

像素。是一个虚拟长度单位（Pixel，简写px），是计算机系统的数字化图像长度单位，如果像素要换算成物理长度，需要指定精度DPI（Dots Per Inch，每英寸像素数），在扫描打印时一般都有DPI可选。Windows系统默认是96dpi，Apple系统默认是72dpi，一般打印时为300dpi。

em。在层叠样式表中，单位em是字体点数或英寸数在名义上的高度。任何指定的字体在实际上、物理上的高度取决于用户定义的DPI设置、当前元素的字体大小以及所用的字体。为使样式规则只取决于默认的字体大小，发展出了另一个单位：rem。rem或root　em是文件中根元素的字体大小。不同于会随着元素的不同而变化的em，rem在文件中是固定的。通常1em＝16px，而16px是浏览器默认的。CSS其他单位见3.5节。

字号、磅、像素、em之间的关系

表14　字号、pt、px、em之间的换算

字号	Pt	px	em	字号	pt	px	em
初号	42	56	3.5	小初	36	48	3
一号	26	35	2.2	小一	24	32	2
二号	22	29	1.8	小二	18	24	1.5
三号	16	22	1.4	小三	15	21	1.3
四号	14	19	1.2	小四	12	16	1
五号	10.5	14	0.875	小五	9	12	0.75
六号	7.5	10	0.625	小六	6.5	8	0.5
七号	5.5	7	0.4375	八号	5	6	0.375

字级和字体的选择原则为和谐、美观。符合统一、一致、有序、不倒置的规则。一部书同一性质的文字只能选择同一种字级、字体，图表及注释文字不能超过正文的字级；各级标题用字的字级，必须符合大小有序的规则。

9.6 标点符号

标点符号是图书内容语句的组成部分，对表达语句的含义和性质有其辅助的作用。排版时正确地使用标点符号的幅面和位置以及注意标点符号的禁排规范，对确保读者的阅读体验尤为重要。多看阅读器在标点的禁排处理的比较好，电子书制作人员在排版时基本不用考虑这类问题。

全角和半角

全角和半角（传统出版上称作对开），是计算机中显示字符的格式。

传统上，英语或拉丁字母语言使用的电脑系统，每一个字母或符号，都是使用一字节的空间（一字节由8比特组成，共256个编码空间）来储存；而汉语、日语及韩语文字，由于数量大大超过256个，故惯常使用两字节来储存一个字符。在使用等宽字体（指字符宽度相同的电脑字体）的环境下，中日韩文字此时占据两倍于西文字符的显示宽度。所以，中、日、韩等文字称为全角字符，相比起来，拉丁字母或数字就称为半角字符。有时为了使字体看起来齐整，英文字母、数字及其他符号也由原来只占一个字空间，改为占用两个字的空间显示、使用两个字节储存的格式。

标点符号的名称和术语

标点符号的名称和术语如表8所示。

表15　标点符号的名称和术语

标点符号	名称	术语	标点符号	名称	术语
。	句号	句圈、脚圈	（　）	括号	
，	逗号		［　］	方括号	硬方括号
、	顿号	顿点	□　□	六角括号	软方括号
；	分号		【　】	黑括号	鱼尾
：	冒号		〈　〉	书名号	
？	问号		《　》	书名号	
！	感叹号	叹号	——	破折号	两字杠
□□	单引号（竖排）	单勾股	……	省略号	三连点、连点
□□	双引号（竖排）	双勾股	·	着重号	重点、加重点

『』	单引号（横排）		专名号	人名线
「」	双引号（横排）	□□	书名号	花线

标点符号的幅面和位置

从20世纪50年代开始，标点符号只有一种幅面（全身）和一种位置（居中）。根据图书内容版式的需要，标点符号现为两种幅面，一种是全身（全角）、一种是半身（对开，半角）。标点符号的位置分为三种，一种是居中，用于横排文和竖排文。一种是左下角，用于横排文。一种是右上角，用于竖排文。

标点符号的制式

全身制。所有标点符号全部占一个汉字的位置，所以叫「全身制」。但当两个标点符号排在一起时，前一个采用对开，以免过于稀疏，适用于理论书和文艺书。

对开制。全部标点符号都排成对开。有些科技书、报纸和工具书为节省版面采用这种形式。

开明制。仅句号、问号、叹号占一个汉字的位置（全身），其他标点符号全部占半个汉字位置（对开）。该制式的幅面贴近语句语法的要求，目前大多数出版物都是采用这种制式。该制式符号在20世纪50年代由上海开明书店首先应用于正文排版，所以称为「开明式」符号。

实际上在电子书中排版并不需这么复杂，只要注意中文书使用全角标点符号，英文书或英文句子、段落使用半角标点符号即可，剩下的标点压缩交给阅读器处理。

标点符号的排版规范

标点符号的排版位置和占字身幅面的宽度，一定要符合排版规范。否则对表达语句含义和阅读将有一定的影响。标点符号的排版规范中有些属于排版引擎的工作，在制作时不需要考虑这方面的问题，另外就是一些特殊情况，比如原始文档在传播过程中全角标点变成了半角，那么就需要根据排版规范修改为全角，这才是我们在做电子书时要注意的问题。

句号。问号？叹号！逗号，顿号、分号；冒号：。不能排在每一行之首，防止出现「顶头点」。相信这一点大部分人在写作文时，语文老师都提示过。

引号「」括号（）书名号《》。以上三种符号前一半不能排在每行的行尾；后一半不能排在每行的行首。

省略号……。不可以排在行首，两个三连点更不能分开。

破折号——。不可以排在行首，两个字长，不能用一个字长的至号组合。

间隔符。用全身的中点，不能用半身的中点。如：用来分开标题中并列的词（水浒·活捉）；用来间隔外国的名、姓（小罗伯特·唐尼）。

书名线、专名线、着重点。书名线、专名线、着重点，横排书应排在文字下方，亦即排在行间，标点下一般不排。如：

> 鲁迅先生的祝福，深刻揭露和鞭挞了封建社会和封建礼教的吃人本质。

横排书槛目前用这三种符号的已比较少，如专名线一般不放，书名线改为「《》」；着重点则多改排黑体。

两个标点符号连排。图书用全身制或者开明制、第一个符号必须用半身幅面，第二个符号最好使用全身幅面。

正文字号段首序号，标号的幅面。段首序号如「（一）」、「（1）」，左右的标号。图书使用任何制式的标点符号，第一个「（」符号用半身幅面，第二个「）」符号用全身幅面；又如「一、」「1.」顿号和尖点也应该用全身幅面。

横排文引号和竖排文引号。横排正文中的单引号和双引号（『』）、（「」），竖排文改排（﹁﹂）、（﹃﹄）。反之也必须改排。

中华人民共和国非强制性国家标准：先用双引号「」，内部如需再引用，再用单引号『』，若再需引用，使用双引号「」，以此类推。竖排仍保持双引号在外，改用﹃﹄和﹁﹂。

台湾教育部：先用单引号「」，内部如需要引用，再用双引号『』。而双引号内部又需要引用，则再用单引号，如此类推。

香港标准是先用单引号「」，内部如需要引用，再用双引号『』，竖排文字则为﹁﹂和﹃﹄。

关于竖排的实现见10.9节。

容易混淆的标点符号。

①千位点和小数点。必须用半身符号，不能用全身。

②冒号和比号。冒号在半身幅面的下部，比号在半身幅面的中部。

③音节号与后单引号。外文中的音节号是四分身，后单引号是二分身。

④书名号与远大于号，远小于号。书名号「《》」，与数学「远大于≫」、「远小于≪」不要混用。

⑤连接符、减号、连字符。连接符占满一个字字身，减号占2/3字身，连字符占半字身。

三种幅面「—」号的排版。半身「-」用于「的」的含义，如NY-A、生态-地理。全身「—」用于「至」的含义，如1999—2000。两个字「——」用于「说明」的含义，如——引自排版与校对规范（第二版）。

外文排版使用的标点符号

同是「，」号中文标点符号和外文标点符号不同，中文标点符号的位置在左下角，而外文标点符号的位置也是在左下角，但必须与「外文基线」平齐（外文基线是指外文大写字母的底部）。因为外文标点符号比中文标点符号位置略高，所以又称为「高脚符号」。在排外文是输入法一定要切换到外文符号状态。

外文标点还有一个特点是句号、逗号、分号、冒号等符号，它们下面的「尖点」对齐。

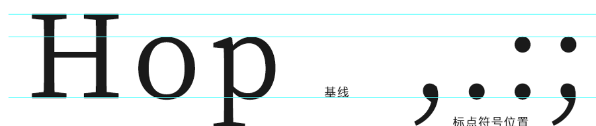


图9-10　外文基线和标点符号位置图

上图中字体为思源宋体，字重为Regular。关于外文「基线」问题可以参看vertical-align属性。

9.7　符号

符号是用象形、图形或外文代表着某一事物、规则和短语的符号。作为排版人员和校对人员，必须要掌握符号的形状和用途。外文符号还需要掌握外文的文种，大写和小写以及正体和斜体。排版时符号使用得是否规范正确，直接影响图书内容的质量。如果符号使用得不正确还影响图书的档次。

各种数字符号

阿拉伯数字。阿拉伯数字基本上分为4种字体，即正体、白斜体、黑正体、黑斜体。常用的数字符号是白正体。黑正体和黑斜体常用于标题字。

白正体　1234567890　白斜体　*1234567890*

黑正体　**1234567890**　黑斜体　***1234567890***

带装饰数字。带装饰的数字符号有两种，一种是圈阳纹数码，一种是圈阴纹数码。阳纹和阴纹又有中文数码和阿拉伯数码之分。一般用于图书内容注释序号。

阿拉伯圈阳码　①②③④⑤⑥⑦⑧⑨⑩

阿拉伯圈阴码　❶❷❸❹❺❻❼❽❾❿

中文圈阳码　□□□□□□□□□□

中文圈阴码　略。

在电子书排版中这些带圈的数字输入不便，很少使用，最常见的是阿拉伯圈阳码。

中文数字。中文数字有两种形式，一种是全程，一种是简称。全程是按口语排列，如三千零四十，简称则按阿拉伯数码形式，如三〇四〇。简称又分为两种幅面，一种是全身，一种是上下二分的二分身。一般用于竖排书的表格、注释序号和中缝页码。

罗马数字。罗马数字是用拉丁文字符组成的数字，可以用大写字母或小写字母表示。

I（1），V（5），X（10），L（50），C（100），D（500），M（1000）

V、X、L、C、D、M六个字符的上边加一条横线，表示该字符代表的数值增值一千倍的含义。但常用数字一般在「20」以下，多用于科技书小标题的序号，正文前的页码内容（一般用小写）。

罗马数字是以「加」或「减」的形式表示数值的大小：字符代表的数字，同级数组合表示代表的数字相加；大小数组合，大数左边的是减数，大数右边的数是加数。

I II III IV V VI VII VIII IX X XI XII L C D M

i ii iii iv v vi vii viii ix x xi xii l c d

数学运算符号

常用符号在排版时不要用错，如比号、冒号、大于号、书名号、和号和积分号的幅面等。在使用正则表达式替换标点符号是，需要特别注意。

表16　常用数学符号

符号	名称	术语	符号	名称	术语
+	加号、正号		⋚	小于大于	
-	减号、负号		≯	不大于	
×、·	乘号	「·」代乘号	≮	不小于	
÷	除号		≫	远大于	
:	比例号	比号	≪	远小于	
/，—	分数号	「/」斜线	～，—	连接号	弯直至号
±	加减号	正负号	∽	相似于	
∓	减加号		∞	无穷大	横八
=	等号		∵	因为	
≡	全等于		∴	所以	
≈、≃、≐	近似于		||	绝对值	
≡	恒等于		‰	千分号	
≠	不等于		√	平方根好	根号
≒	约等于		⌀	直径	直径号
>	大于		Σ	集和号	
<	小于		Π	连乘号	
≥、≧	大于或等于		∫	积分号	
≤、≦	小于或等于		(	括弧、小括弧	圆括号
⋚	小于等于或大于		[	括带、中括弧	方括号
⋛	大于等于或小于		{	括号、大括弧	大括号
≷	大于小于				

∀∁∂∃∄∅∆∇∈∉∊∋∌∍∎∏

∐∑−∓∔∕∖∗∘∙√∛∜∝∞∟

∠∡∢∣∤∥∦∧∨∩∪∫∬∭∮∯

∰∱∲∳∴∵∶∷∸∹∺∻∼∽∾∿

≀≁≂≃≄≅≆≇≈≉≊≋≌≍≎≏

≐≑≒≓≔≕≖≗≘≙≚≛≜≝≞≟

≠≡≢≣≤≥≦≧≨≩≪≫≬≭≮≯

≰≱≲≳≴≵≶≷≸≹≺≻≼≽≾≿

⊀⊁⊂⊃⊄⊅⊆⊇⊈⊉⊊⊋⊌⊍⊎⊏

⊐⊑⊒⊓⊔⊕⊖⊗⊘⊙⊚⊛⊜⊝⊞⊟

⊠⊡⊢⊣⊤⊥⊦⊧⊨⊩⊪⊫⊬⊭⊮⊯

⊰⊱⊲⊳⊴⊵⊶⊷⊸⊹⊺⊻⊼⊽⊾⊿

⋀⋁⋂⋃⋄⋅⋆⋇⋈⋉⋊⋋⋌⋍⋎⋏

⋐⋑⋒⋓⋔⋕⋖⋗⋘⋙⋚⋛⋜⋝⋞⋟

⋠⋡⋢⋣⋤⋥⋦⋧⋨⋩⋪⋫⋬⋭⋮⋯

⋰⋱⋲⋳⋴⋵⋶⋷⋸⋹⋺⋻⋼⋽⋾⋿

以上各种符号的名称见Unicode®。

外文字母

作为一名与出版专业有关的人员，还必须掌握外文字母的语种；大写字母与小写字母的差别；正体字母与斜体字母的差别，以适应科技书排版的需要。

表17　拉丁文字母表

印刷体	正体	Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz
	斜体	*Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz*
手写体		*Aa Bb Cc Dd Ee Ff*

		Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz

演示的字体名称为Times New Roman。

表18　德文字母表

印刷体	正体	Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz
	斜体	*Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz*
哥特体		Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz

德文花体又称哥特体，正体和斜体为Times New Roman，哥特体名称为OldLondon。

表19　俄文字母表

印刷体	正体	Аа Бб Вв Гг Дд Ее Ёё Жж Зз Ии Йй Кк Лл Мм Нн Оо Пп Рр Сс Тт Уу Фф Хх Цц Чч Шш Щщ ъ ы ь Ээ Юю Яя
	斜体	*Аа Бб Вв Гг Дд Ее Ёё Жж Зз Ии Йй Кк Лл Мм Нн Оо Пп Рр Сс Тт*

Уу Фф Хх Цц Чч Шш
Щщ ъ ы ь Ээ Юю Яя

演示的字体名称为Times New Roman。

表20　希腊文字母表

印刷体	正体	Αα Ββ Γγ Δδ Εε Ζζ Ηη Θθ Ιι Κκ Λλ Μμ Νν Ξξ Οο Ππ Ρρ Σσ Ττ Υυ Φφ Χχ Ψψ Ωω
	斜体	*Αα Ββ Γγ Δδ Εε Ζζ Ηη Θθ Ιι Κκ Λλ Μμ Νν Ξξ Οο Ππ Ρρ Σσ Ττ Υυ Φφ Χχ Ψψ Ωω*

演示的字体名称为Times New Roman。

正体和斜体的外文符号

外文符号就是用外文字符作为符号的代表。外文符号是科技类书刊的重要组成部分。外文符号的大写和小写，正体和斜体，在排版时排得是否正确，直接影响图书的质量，严重的有可能曲解了作者的原意。

外文符号正体和斜体排版使用的原则。科技书外文符号排版的原则基本上有两条：凡是外文符号代表数字含义的用斜体排版；凡是外文符号代表单位、人名、缩写词、简短语用正体排版。为了更好地掌握原则，还必须熟知外文常用符号的名称和作用。

外文正体的常用符号。

①数学公式中的简写词。用小写正体排版。小写简写词出现在数学公式算是当中，不易辨认。用正体的小写简写词一般是「词的字符连写」。常用词见下表。

表21　常用术语简写词表

简写词	名称	简写词	名称	简写词	名称
log	对数	arc tg	反正切	mod	模数
lg	十进对数	arc ctg	反余切	sn. cn. dn	椭圆符号
ln	自然对数	arc	反	lim	极限
sin	正弦	sh	双曲正弦	const	常数、恒量
cos	余弦	ch	双曲余弦	ind	指数
tg	正切	th	双曲正切	max	极大、最大
ctg	余切	cth	双曲余切	min	极小、最小
sec	正割	arsh	反双曲正弦	np	投影
csc	余割	arch	反双曲余弦	div	散度

arc sin	反正弦	arth	反双曲正切	cur. rot	旋度
arc cos	反余弦	arcth	反双曲余切	grad	梯度、陡度

②常用的单位名称。用正体排版，有的单位名称用大写外文，有的单位名称用小写外文。排版时要按规范使用不能混淆。常用单位名称和外文符号见下表。

表22　常用术语简写词表

符号	单位名称	符号	单位名称	符号	单位名称
μm	微米（微）	L（l）	升	kVA	千伏安
mm	毫米	kL	千升	Ω	欧姆
cm	厘米	mL	毫升	min	分
m	米	Å	埃（长度）	h	小时
km	千米	F	法拉（电容量）	dB	分贝
mg	毫克	R	电阻	π	圆周率
g	克	C	电容	HP	马力
kg	千克	kHz	千赫兹	Ph	辐透（照度）
t	吨	kW	千瓦	pH	酸碱度

代表单位字符前的代表量字符的含义是μ（微），m（毫），c（厘），d（分），k（千）。

③国家标准、产品型号、材料代号。国家标准是国家制定产品的标准，产品型号是单位厂家制定的标准。每个字符都代表着产品标准的说明，一般用大写正体排版。产品标准与产品分类中间的「-」号，必须用半身排版。代表材料的字符是元素符号，每个元素用大写或小写排版。

④代表形状的符号。又称为形号，用字符的形状指定某种物质的形状用大写正体排版。如T形楼房，O形体育场，工字钢材，U形花坛等。

⑤表示一般标量的算符。用正体排版，该字符总是以排比的形式出现在数学公式中。如f(x)f(y)(z)，∂x ∂y ∂z，dx dy dz等。

⑥国名的缩写。如P.R.C. 中华人民共和国。U.S.A. 美利坚合众国。

外文斜体的常用符号。在科技书中凡是用于代表「数」的外文符号基本上都用斜体排版。因为斜体与手写体近似，所以排版采用斜体，另外与单位符号区别开，公式显得条理清楚，便于阅读。

如代表已知数的a，b，c；代表未知数的x，y，z。

9.8 装饰符号

装饰符号是除文字和图像以外在刊物或报纸上用于编排版面和美化版面的符号。包括线、边和底纹三种形式。

线

CSS中可以使用border-style创建4种常用的边框：点状（dotted）、实线（solid）、双线（double）、虚线（dashed）。见3.4.3节。

边

边也是线条的一种形式，可以通过border-image属性控制，见3.4.3节。

「边」一般用于图书中较大段的文章和一个版面的围框。

底纹

用于图书内容背景的花纹称为「底纹」。底纹可以通过background属性控制，见3.4.2节。

选用底纹的式样必须要考虑底纹的用途，花样选用得当，起到衬托图书内容的作用。另外，在做整段文章的底纹时，底纹要选用浅色的，避免文字看不清。

9.9　开本

开本是图书幅面大小的单位；开本也是说明图书大小的尺寸。

开本

开本是以全张纸幅面的大小为准，开（裁）多少张同样大小幅面的纸即为多少开的开本（开数）。由于纸面的幅度不一样，同样开本的尺寸是不一样的。

16开与A4

16开、32开是我国传统的称谓，如前所述，16开是全张纸开出16张同样大的纸。A4是国际系列纸张大小的称谓。A4表示A系列大纸对折四次（裁切四次）。B4则表示B系列大纸对折四次。A5或B5依次类推。总之，开指的是裁切数量；A4中的「4」指的是裁切次数。

正常开本的开法。全张纸长边对折依次向小幅面开。两个相邻的开本，幅面形式相同，幅面大小相差一倍。正常开本长和宽的比例是1：√2（1：1.414）。

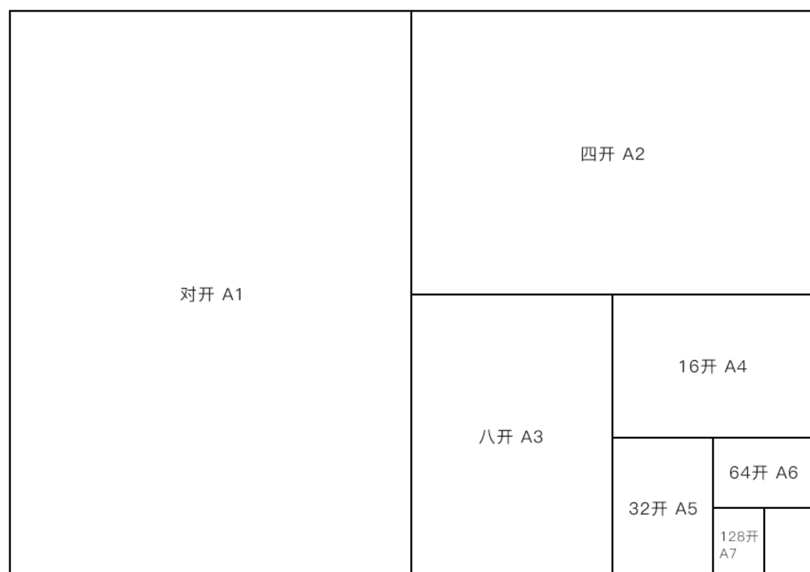


图9-11　正常开本示意图

全张纸和开本的尺寸

目前我国图书用纸有两大系统，一种是我国传统标准系列（787mm×1092mm，850mm×1168mm）；一种是国际标准系列（A系列880mm×1230mm和900mm×1280mm，后者是出血图和封面用纸；B系列1000mm×1400mm）。A系列880mm成1240mm的幅面约等于1㎡。

常用的版心尺寸

版心大小可以分为三大类，一种是理论、文艺图书，一种是科技图书，再一种是工具图书，科技树的版心在版面中占有率比较适中；理论、文艺书版心略小；工具书版心略大。一般书刊的版心尺寸见下表（成书尺寸、版心尺寸单位为毫米mm）：

表23　书刊成书和版心尺寸

开本	成书	字号	版心
16开	184×260	五	147×219
大32开	140×203	五	103×158
140×203	小五	104×159	
32开	130×184	五	74×108
130×184	小五	101×150	
大64开	101×137	五	74×108
101×140	小五	73×107	
64开	92×126	五	70×103
92×130	小五	70×102	

通常如果是用于装订，会在成书的四边各加3mm的出血。通过成书尺寸减去版心尺寸，即可算出天头地脚、订口切口的宽度。

9.10　一般图书内容的顺序

图书的组成指构成图书的要素，以及区别于非书形式出版物的一些重要特征，包括构成一本书的各个部分的内容，总的分为正文和辅文两大部分。正文为一本书的主体，是著作者提供给读者的主要知识内容，常有篇、章、节、目等一定格式体系。正文之外的所有辅助文字统称辅文，包括作者简介、内容提要、版权说明、出版说明、前言、目次、凡例、附录、后记、参考文献等。

纸书的逻辑结构一般指的是内容部分，它们一般有一个约定俗成的逻辑顺序。下面列表展示了一本书的一般结构：

- 封面（Front cover）
- 卷首的空页（Front endpaper）
- 折口（Flap）
- 衬页（Flyleaf）
- 前页（正文前的书页；Front matter）
    - 卷头插画（Frontispiece）
    - 标题页（Title page）
    - 扉页（Colophon）
    - 版权页（Copyright page）
    - 内容目录（Table of contents）
    - 图片列表（List of figures）
    - 表格列表（List of tables）
    - 献辞（Dedication）
    - 致谢（Acknowledgments）
    - 前言（Foreword）
    - 引语（Preface）
    - 简介（Introduction）
- 正文（Body）
- 末尾的书页（Back matter）
    - 附录（Appendix）

    - 词汇表（Glossary）
    - 索引（Index）
    - 注释（Notes）
    - 参考书目（Bibliography）
    - 版权页标记（Colophon）
- 衬页（Flyleaf）
- 卷尾的空页（Rear endpaper）
- 封底（Rear cover）

上面的顺序是一般图书的结构顺序，具体到个例时可能只包含上面的部分。

序与前言

序通常置于书名页后的首页上，序在不同类型的书稿中可称为序言、前言、绪言、叙言、弁言、序文、引言、小引等，有时也写为「写在前面」、「作者的话」、「编者的话」或「致读者」等。

作者或出版者请他人写的序为他序，他人写的文章选来作序的称为代序；作者本人写的序为自序。序还有多卷书的总序与本册书的分序、初版序与再版序、原版序与译序之分。

序的内容通常包括：

1. 本书主要内容特色及表现形式方面的特点；
2. 写作背景和写作过程，与本书内容相关的作者经历及业绩；
3. 介绍或探讨本书的学术价值及现实意义；
4. 对读者如何阅读、使用本书的建议；
5. 借序发表有关理论见解、学术思想；
6. 借序回答读者关心的问题；
7. 介绍对本书作出贡献和帮助的人或事；
8. 翻译书的有关情况说明等。

各种序并存情况下的排序惯例如下：

①译者序→②丛书序→③再版序→④代序、他序→⑤自序

凡例

凡例是书籍辅文的一种文体名，又称体例、例言，是说明书籍编纂体例的文字，多用于地图册、科技专著和辞书、手册等工具书。

凡例通常向读者说明选文标准或收词原则、归类方式与编排顺序、译义范围与释文要求、行文规范、检索方法以及辅文安排，目的是方便读者阅读和检索。

题解

题解是说明书籍或其某些篇章的作者、内容、版本等情况的文字，是书籍辅文的一种文体。

题解在书名或篇章名之后，少者几个字，多则数百字，主要用来：

1. 解释标题中出现的词语，如读者比较生疏的人名、地名、词牌名等；
2. 说明作品的版本情况，如选自何书何处，出版时间、地点、单位；
3. 作品简介及背景资料；
4. 对作品的评价等。

题解常见于历史文化名著的近、现代版本。

编者的话

编者的话是编者对入选作品所作的说明或评论文字，也称编者按语，是辅文的一种文体名。编者的话主要对作品所涉及的人物、事件、观点、方法、数据等进行说明、分析和评论，目的在于引起读者对某些问题的重视，从而深入学习、思考、研究。

编者按对入选作品或褒或贬，发人深省，常置于文章标题之上方的醒目位置。编者的话表明了编者、出版者的观点和立场，文字虽短，关系重大，必须经过三审，由终审者定稿。

注释

注释是对正文中的文字、读音、词汇、典故、史实、引文出处或内容意义方面所作的说明文字，是书籍辅文的一种文体。

注释有诂、训、传、说、记、解、笺、章句、集解、义疏、正义等不同名目，或分为原注（原作者自注）、译者注、编者注、校者注等，可编排夹印于正文中间为夹注，或排印于页末为脚注，或作其他安排。

注释是作品的组成部分，反映了注文作者的政治立场、思想观点、治学态度和学术水平，要力求准确无误，编辑和作者要像审读正文一样对注释内容认真审读核查。

后记

后记又称跋、编后语、附记等，是书籍辅文的一种文体名。后记是附于作品末，用以介绍、评析作品的说明性文字，与「前言」、「序」一样具有评介作品、指导阅读的作用，但侧重于介绍成书过程中的有意义的事情和问题、感受，可用来进一步介绍作者、作品的思想基础，表述写作的认识、体会、感想以及对读者的希望。

「后记」、「跋」一般由作者本人撰写，「编后」由主编者或编辑撰写。

校订者也可写「校后记」、译者也可写「译后记」。

后记等不是作品的必有内容，由编辑、作者根据实际需要决定写或不写。

参考文献

参考文献或参考书目，是书籍常用辅文之一，指作者为创作、撰稿需要，所参考、查阅、适当引用过的他人著作、论文或其他有关文献资料。

为尊重他人著作权，也为了说明作品有关内容、观点和有关事实的依据与来源，便于读者查考、研讨，作者有责任在文中或文后列出参考文献。

索引

索引是为方便读者按某方面需要查阅书中某类事物、概念等而依一定顺序排列，注明所在页码，附于正文后的检索工具。索引是书籍辅文之一。

索引的种类有内容索引、名词术语索引、中外人名索引、地名索引、重大事件索引、中外文对照索引、年代或朝代索引、图题或表题索引、主题索引、文献索引等。索引的编排可按汉语拼音或外文字母音序、汉字笔画或部首的形序、主题类序、年代或朝代的时序或数字的顺序等方式排列。

索引的编制一般由作者承担，编辑核查确定。索引一般放在书末，与正文连续编页码，并编入全书目录之中。某本书如需要，可以多种索引并存。

出版说明

出版说明又称出版者的话，以编辑部或出版社的名义对组织编辑、出版该书所作的说明，是书籍辅文的文体之一，其内容侧重于介绍组稿与出版目的、与作者商讨编写方案的情况、编辑审读书稿与编辑加工的情况、对书中一些疑难问题的处理情况等出版者认为有必要让读者了解的事项，其目的也是为读者深入理解书中内容提供必要的信息。出版说明一般不对作者或作品作评价。

出版说明置于正文前、后均可，多置于目录前。

附录

附录是附于正文后面方便读者查阅的各种文字或图表资料，以及与正文内容密切相关又不适合写入正文的文章、文件、地图、表格等资料。

附录是为正文服务的，也是为读者学习、应用书中内容提供服务的。

正确选取附录内容，往往可以提升书籍的使用价值。

内容提要

内容提要又称内容简介、内容说明、内容摘要等，是书籍辅文的一种文体，是介绍书籍主要内容及其特色的概括性文字。

内容提要一般置于版本记录页的上方，也可安排在封面的勒口、封二或封底上。内容提要是出版者向读者或销售者介绍、推荐该书的文字材料，导购作用强。一般读者拿起图书观赏该书的封面、封底后首先阅读的文字就是该书的内容提要。

内容提要一般由编辑撰写，或者由作者写初稿，再经编辑修改定稿。

写好内容提要是编辑的责任，要努力提高其质量、水平。内容提要应文字简洁、准确可靠、突出重点，除了对著作背景或作者情况作简要介绍外，还可根据不同书类分别写成评介型、说明型、梗概型、悬念型短文，以激发读者的兴趣。

9.11 字符编码

在计算机中，所有的数据在存储和运算时都要使用二进制数表示，例如：a、b、c、d这样的52个字母（包括大写）、以及0、1等数字还有一些常用的符号（例如*、#、@等）在计算机中存储时也要使用二进制数来表示，而具体用哪些二进制数字表示哪个符号，当然每个人都可以约定自己的一套方法（这就叫编码），大家如果要想互相通信而不造成混乱，那么就必须使用相同的编码规则，于是美国有关的标准化组织就出台了ASCII编码，统一规定了上述常用符号用哪些二进制数来表示。

GB 2312

后来计算机传到中国之后，为了解决汉字显示的问题，有了GB2312标准，中文名《信息交换用汉字编码字符集》。GB2312编码适用于汉字处理、汉字通信等系统之间的信息交换，通行于中国大陆；新加坡等地也采用此编码。中国大陆几乎所有的中文系统和国际化的软件都支持GB2312。GB2312标准共收录6763个汉字，其中一级汉字3755个，二级汉字3008个；同时，GB2312收录了包括拉丁字母、希腊字母、日文平假名及片假名字母、俄语西里尔字母在内的682个全角字符。整个字符集分成94个区，每区有94个位。每个区位上只有一个字符，因此可用所在的区和位来对汉字进行编码，称为区位码。

GB2312的出现，基本满足了汉字的计算机处理需要，它所收录的汉字已经覆盖中国大陆99.75%的使用频率。对于人名、古汉语等方面出现的罕用字，GB2312不能处理，这导致了后来GBK及GB18030汉字字符集的出现。

GBK

GBK全称《汉字内码扩展规范》（GBK即「国标」、「扩展」汉语拼音的第一个字母，英文名称：Chinese Internal Code Specification），中华人民共和国全国信息技术标准化技术委员会1995年12月1日制订，国家技术监督局标准化司、电子工业部科技与质量监督司在1995年12月

15日将它确定为技术规范指导性文件。这一版的GBK规范为1.0版。GBK向下与GB2312编码兼容，向上支持ISO/IEC10646-1国际标准，是前者向后者过渡过程中的一个承上启下的产物。

GBK编码是在GB2312标准基础上的内码扩展规范，使用了双字节编码方案，其编码范围从8140至FEFE（剔除xx7F），共23940个码位，共收录了21003个汉字，完全兼容GB2312标准，支持国际标准ISO/IEC 10646和国家标准GB13000中的全部中日韩汉字，并包含了BIG5编码中的所有汉字。

GB18030

国家标准GB18030-2005《信息技术中文编码字符集》是我国继GB2312-1980和GB13000-1993之后最重要的汉字编码标准，是我国计算机系统必须遵循的基础性标准之一。GB18030有两个版本：GB18030-2000和GB18030-2005。GB18030-2000是GBK的取代版本，它的主要特点是在GBK基础上增加了CJK统一汉字扩充A的汉字。GB18030-2005的主要特点是在GB18030-2000基础上增加了CJK统一汉字扩充B的汉字。

GB18030-2005《信息技术中文编码字符集》是我国自主研制的以汉字为主并包含多种我国少数民族文字（如藏、蒙古、傣、彝、朝鲜、维吾尔文等）的超大型中文编码字符集强制性标准，其中收入汉字70000余个。

Unicode

Unicode（万国码、国际码、统一码、单一码）是计算机科学领域里的一项业界标准。它对世界上大部分的文字系统进行了整理、编码，使得电脑可以用更为简单的方式来呈现和处理文字。

Unicode伴随着通用字符集的标准而发展，同时也以书本的形式对外发表。Unicode至今仍在不断增修，每个新版本都加入更多新的字符。Unicode涵盖的数据除了视觉上的字型、编码方法、标准的字符编码外，还包含了字符特性，如大小写字母、Emoji等。可在下面的链接中查看Unicode的最新版本。

通用规范汉字表

《通用规范汉字表》是由中华人民共和国教育部、国家语言文字工作委员会联合组织研制的汉字使用规范，自2001年开始研制，原定名《规范汉字表》，历时八年后于2009年8月12日发表了征求意见稿。该字表整合了《第一批异体字整理表》（1955年）、《简化字总表》（1964年初发表，最后修订于1986年）、《现代汉语常用字表》（1988年）以

及《现代汉语通用字表》（1988年），并根据中国大陆用字现状加以修补和完善。《通用规范汉字表》于2013年6月5日正式颁布，成为社会一般应用领域的汉字规范，原有相关字表从即日起停止使用。

字表共收字8,105个，其中一级字表（常用字集）3,500个，二级字表3,000个，三级字表1,605个。一、二级字表主要满足出版印刷、辞书编纂和信息处理等方面的一般用字需要；由姓氏人名、地名、科技术语和中小学教材常见文言文用字构成的三级字表则主要满足与大众生活密切相关的专门领域的用字需要需要。

为尊重大众用字习惯，《规范汉字表》恢复了45个主要用于人名和地名异体字，例如《瑷珲条约》的「珲」字等（吉林省珲春县一直使用这个字作为县名）。

《通用规范汉字表》中的汉字并不是都在基本区内，一些类推简化的汉字实际在扩展区内。

关于Unicode字符可以参考Unicode®字符百科网站。

Unicode在很长一段时间内无法推广，直到互联网的出现。为解决Unicode如何在网络上传输的问题，于是面向传输的众多UTF（UCS Transfer Format）标准出现了。UTF-8是每次8个位传输数据，而UTF-16是每次16个位。UTF-8是在互联网上使用最广的一种Unicode的实现方式，这是为传输而设计的编码，并使编码无国界，这样就可以显示全世界上所有文化的字符了。UTF-8最大的一个特点，就是它是一种变长的编码方式。它可以使用1~4个字节表示一个符号，根据不同的符号而变化字节长度，当字符在ASCII码的范围时，就用一个字节表示，保留了ASCII字符一个字节的编码做为它的一部分，注意的是Unicode一个中文字符占2个字节，而UTF-8一个中文字符占3个字节）。从Unicode到UTF-8并不是直接的对应，而是要过一些算法和规则来转换。

因为传输编码的原因，在EmEditor和Sigil中使用的汉字正则表达式写法不同，见4.3节。

中日韩统一表意文字

中日韩统一表意文字（CJK Unified Ideographs），也称统一汉字（Unihan），目的是要把分别来自中文、日文、韩文、越南文、壮文中，起源相同、本义相同、形状一样或稍异的表意文字，赋予其在ISO/IEC 10646及Unicode标准中相同编码。

比如一般汉字用`[\u4e00-\u9fa5]+`即可匹配到，不过这个正则表达式对「火星文」就不太适用了，甚至全角的标点符号都不包含在内，这时候你就需要更强大的正则表达式了。使用`[\u2E80-\uFE4F]+`正则表达式基本

都涵盖了所有常见字符。

最初期的统一汉字共20,902字，其范围为`U+4E00-U+9FA5`。

扩展A区包含有6,582个汉字，位置在`U+3400-U+4DB5`。

扩展B区包含有42,711个汉字，位置在`U+20000-U+2A6D6`。

于2009年10月发布的Unicode 5.2涵盖了扩展C区，共收录4,149个汉字，包括来自中国大陆、澳门、台湾、日本、越南等尚未被编码的汉字。位置在`U+2A700-U+2B734`。

扩展D区包含的都是所谓的「急用汉字」，合共222个新汉字，于2010年下旬发布的Unicode 6.0中，编码范围为`U+2B740-U+2B81F`（实际有字符为`U+2B740-U+2B81D`）。

扩展D区原本计划放置扩展C区未收录的16,000多个汉字，但在2007年5月，台湾撤消了6,545个第二部分字集内私用汉字，不再使用，原因是那些人名用字的拥有人或已去世或已移居外地，此后扩展D区缩减到大约10,000字左右。由于各种阻碍，协议先把数量较少，又急切要收录的汉字提交出来，就是「急用汉字」，以便和统一码6.0版一起发表。提出的急用汉字只有222字（本来有223字，但日本撤回其中一字）。现在文字小组把第二部分字集延后到扩充E区。

扩展E区在2015年6月17日的Unicode 8.0中发布，放置于编码范围`U+2B820-U+2CEAF`。扩展E区本应包含扩展D区中未收录的10,000多个汉字，但在2008年11月，中国大陆以「难以逐个找证据」为理由，撤销了3,215个汉字，这些汉字主要用于地名、人名、姓氏，亦有上千个《中国大百科全书》中的文字。这是继台湾撤销6,545字之后的又一次大规模撤销。之后又经过长时间的检查处理，E区最终定稿，共有5,762字。

「急用汉字」是扩展E区整理后期，各地新发现并急于使用，又等不及放入扩展F区的字；和扩展E区一起收入Unicode 8.0，位置在`U+9FCD-U+9FD5`；其中中国在此处申请收入三字，连同扩展E区的字，通用规范汉字表的8,105字至此全部收入。

扩展F区在2017年6月20日的Unicode 10.0中发布，编码范围为`U+2CEB0-U+2EBEF`。扩展F区来源于新提交的一批汉字，主要包括一千多个方块壮字及数千个佛经、古籍中的用字以及日本户籍用字，共有7,473字。

9.12　中文字体英文名称

本文内容来自　@张鑫旭　整理的文章，这里仅列出常见的几种字体，更多详细内容请阅读原文。

虽然一些常见中文字体，例如「宋体」，「微软雅黑」等，直接使用中文名称作为`font-family`的属性值也能生效，但我们一般都不使用中文名称，而是使用英文名称，主要是规避乱码的风险。还有一些中文字体，直接使用中文名称作为`font-family`的属性值是没有效果的，如「思源黑体」，「兰亭黑体」等，需要使用对应的英文字体名称才可以。

表24　Windows与OS X系统的默认字体

Windows常见内置中文字体		**OS X**常见内置中文字体	
字体中文名	字体英文名	字体中文名	字体英文名
宋体	SimSun	华文宋体	STSong
黑体	SimHei	华文黑体	STHeiti
楷体	KaiTi	华文楷体	STKaiti
仿宋	FangSong	华文仿宋	STFangsong
新宋体	NSimSun	华文中宋	STZhongsong
微软雅黑	Microsoft Yahei	华文新魏	STXinwei
細明體	MingLiU	苹方	PingFang SC

虽然列表中的字体，有的在手机中会显示为系统默认字体，不过我们的制作主要还是在电脑里，了解这些就可以在Sigil中编辑时就看到实际的字体显示效果。另外多看阅读器中的可调用字体，见10.11.5节；掌阅阅读器的可调用字体，请参考附录四。

Kindle电纸书中的内置字体，如下表：

表25　Kindle的内置字体名称

字体名	调用字体名
宋体	STSong
黑体	STHeiti
楷体	STKai
圆体	STYuan
宋體（繁体）	Song T
黑體（繁体）	MYing Hei T
Amazon Ember（衬线）	Amazon Ember

Helvetica（无衬线）	Helvetica
请参考亚马逊内置字体清单，了解更多字体名称。	

9.13 颜色的选择与搭配

丰富的颜色可以活跃版面，让阅读体验更好。因此必须了解关于颜色的一些基础知识。本节内容来自《平面排版基础的基础》第5章。下面列表中是4个常用的关于颜色的网站，这里列出来，以供参考。

1. 中国色
2. 日本の伝統色
3. CSS颜色名
4. 色轮|色彩配置

9.13.1 颜色的种类

世间万物都有颜色，照片和插图也不例外。我们在排版时必须要考虑到「颜色」的因素，并了解和用好它们。

颜色的基础知识

「颜色」指的是反射在物体上的光进入眼睛后的波长，它以紫、蓝、绿、黄绿、黄、橙、红这彩虹般的连续结构（向量）放射，大脑就是通过分析这些波长的长短来识别各种颜色的。自然界中存在着无数种颜色，主要通过「光的三原色（RGB）」和「色的三原色（CMY）」两种方式最大限度地再现着它们。

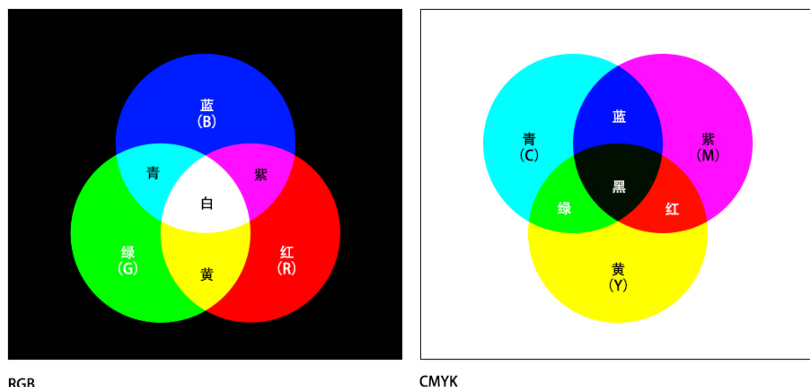


图9-12　RGB和CMYK色彩叠加

上面左图显示的是光的三原色。以红（Red）、绿（Green）、蓝（Blue）三种光的颜色为基础进行各种组合。我们平时用的电脑、手机显示器就是以这种原理来再现颜色的。

右图显示的是色的三原色。通过青（Cyan）、洋红（Magenta）、黄（Yellow）这三种颜色的组合得到各种颜色，通常用于颜料和印刷。将这三种颜色重叠可以得到近似于黑的效果，但很难再现纯黑色，因此在印刷中通常还要用到黑墨水（Black）。

饱和度

饱和度是颜色「鲜艳程度」的标准。饱和度越高越接近原色（最高为纯色），反过来，最低则为无色（黑、白、灰）。

以单色及其深浅来表现图片和照片的单色调（MONOQLO），其实就是将它们的色彩饱和度调到最低的状态。

色相

色相指的是红、黄、绿、蓝、紫这样的色调差异，而所有的色调都是通过颜料或光的组合得以模拟再现的。

在电脑显示中RGB各自分为0～255这256级，能再现256的3次方，约为1678万种颜色。比如要表现「纯蓝」的颜色，就可以将数值设定为R0、G0、B255.它的原理是以简便的数值来尽量模拟自然界中千变万化的颜色。

CMY三原色各自的最大值为100，比如要表现「纯蓝」的颜色，就可以将数值设定为C92%、M75%、Y0、K0。

明度

表现颜色明暗的单位叫明度，明度100%即为白色，0%为黑色。

我们可以用HSB色彩模式描述这三个纬度：H（Hue）代表色相，S（Saturation）代表饱和度，B（Brightness）代表明度。

色相：指色彩的相貌



明度：指色彩的明暗程度



饱和度：指色彩的鲜艳程度



图9-13　HSB示意

常用色

CMYK几乎可以组合出所有色彩，但在电子书中我们一般使用十六进制颜色码，下面给出一些常用色：

大红 #e60012	深红 #b81c22	朱红 #e94829	鲑肉色 #f29b87	桃红 #f19ec2	樱粉色 #fef7f2
橙色 #ed6c00	橘色 #f39800	赭石 #bb5535	棕色 #6a4115	栗色 #956f29	深棕 #533c32
黄色 #ffd900	柠檬黄 #fff33f	金黄 #fabe00	乳黄 #fff3c3	肉色 #fbd8ac	土黄 #c88c2d
绿色 #009944	草绿 #8dc556	茶绿 #598f35	深绿 #00693e	翠绿 #008087	孔雀绿 #0072a6
天蓝 #00a0da	淡蓝 #54c3f1	青绿 #00aebb	蓝色 #0068b7	群青 #1e2678	藏青 #00194b
蓝紫 #4d4398	紫色 #9070af	紫红 #92308d	深紫 #4c0061	灰色 #727171	黑灰 #55554d

9.13.2　颜色的印象

颜色会给我们温暖、冰冷、艳丽、朴素等各种音响，我们要灵活运用颜色的印象来进行排版。

对立的印象

颜色都有各自的印象，比如我们一般认为红色是暖的，蓝色是冷的。人们会以这种方式从颜色产生联想或受到心理影响。在无数种色彩中寻找要用的颜色时不应盲目选择，而要参考颜色所拥有的印象。

下面几组印象对立的颜色。在「强硬VS柔和」中，饱和度和纯度高的颜色显得强硬，饱和度低的颜色显得柔和。在「热闹VS安静」中，接近红色、橙色、黄色的颜色会给人热闹的印象，而接近蓝色和紫色的颜色则给人安静的感觉。在排版时我们要顾及颜色的印象来加以选择。

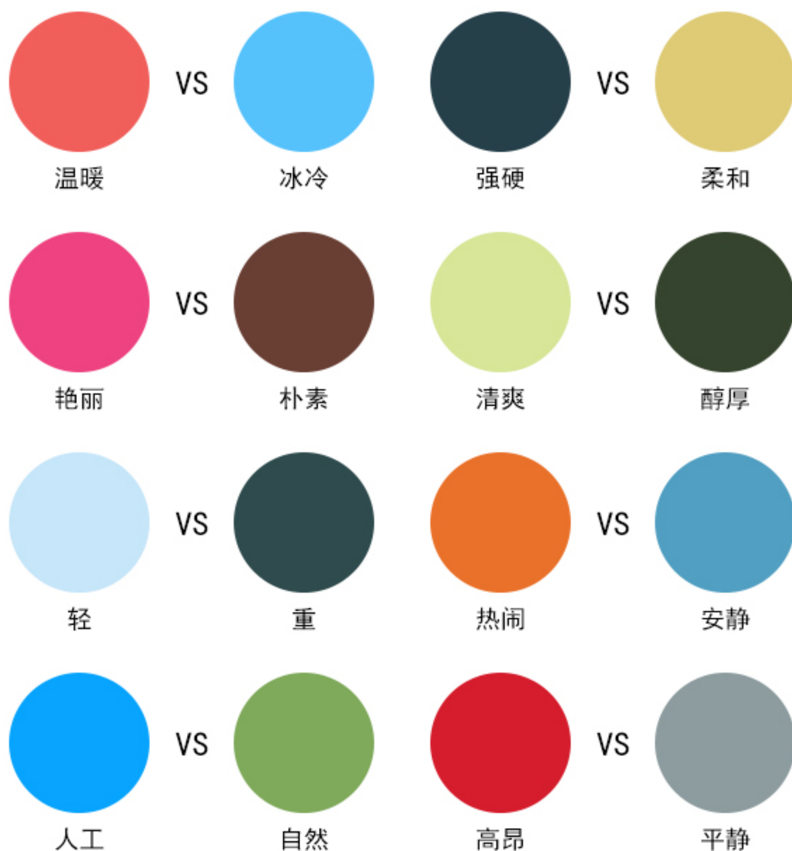


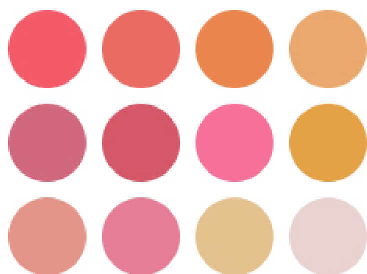
图9-14　颜色的对立印象

暖色调和冷色调

不同的人、不同的国家对颜色的认知也会存在一些差异，不过，对「暖色调」和「冷色调」的概念大致上是有共识的。

暖色调指的是以红色系为中心色相的颜色，冷色调则是以蓝色系为中心色相的颜色。人们认为这是源于暖色调给人太阳和火焰的印象，冷色调则让人联想到水和冰。

从心理学角度来说，暖色调能使人情绪高涨、增进食欲，而冷色调则有镇静、抑制食欲的效果。



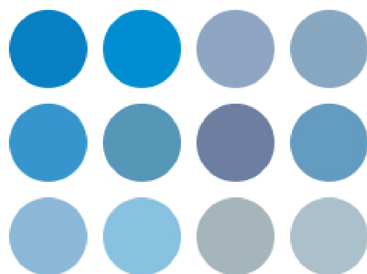


图9-15　暖色调和冷色调

艳丽与朴素

怎样的颜色能给人艳丽或朴素的印象呢？通常，眼里的颜色饱和度和明度较高，较接近原色，而朴素的颜色饱和度和明度较低。

通过比较艳丽和朴素的颜色色相可以发现，暖色调显得艳丽，冷色调显得朴素。





图9-16　艳丽与朴素的颜色

9.13.3　底色与文字颜色

有时在正文排版中也要加入颜色，此时该如何选择底色和文字颜色才能使排版易读又给人留下深刻的印象呢？

底色浅、文字深

正文的必要条件是易读，该如何配色呢？白底黑字是阅读最轻松的搭配，原因就在于色彩明度的差异。下面哪种明度没有显著差异的底色和

文字读起来会十分吃力。

说花美就会有人说「也有不美的花」。预想到会有这种抱怨，于是写：「既有美丽的花也有不美的花」。
这基本是废话。
让所有人都认同的文字称不上表达。

底色：#eaf6fd / 文字颜色：#eeda00

现在来介绍几种明度差异显著的彩色文字排版。背景颜色浅、文字颜色深的排版更易读且更能传递出颜色的印象。我们在正文中加入颜色时，一般会选择浅底、深字的配色，请记住这一点。

说花美就会有人说「也有不美的花」。预想到会有这种抱怨，于是写：「既有美丽的花也有不美的花」。
这基本是废话。
让所有人都认同的文字称不上表达。

底色：#f7f5f1 / 文字颜色：#221815

说花美就会有人说「也有不美的花」。预想到会有这种抱怨，于是写：「既有美丽的花也有不美的花」。
这基本是废话。
让所有人都认同的文字称不上表达。

底色：#f7f5ea / 文字颜色：#7d6b48

说花美就会有人说「也有不美的花」。预想到会有这种抱怨，于是写：「既有美丽的花也有不美的花」。
这基本是废话。
让所有人都认同的文字称不上表达。

底色：#fff67f / 文字颜色：#221815

说花美就会有人说「也有不美的花」。预想到会有这种抱怨，于是写：「既有美丽的花也有不美的花」。
这基本是废话。
让所有人都认同的文字称不上表达。

底色：#f8c5ac / 文字颜色：#221815

说花美就会有人说「也有不美的花」。预想到会有这种抱怨，于是写：「既有美丽的花也有不美的花」。

这基本是废话。
让所有人都认同的文字称不上表达。

底色：#eaf6fd / 文字颜色：#007a8e

说花美就会有人说「也有不美的花」。预想到会有这种抱怨，于是写：「既有美丽的花也有不美的花」。
这基本是废话。
让所有人都认同的文字称不上表达。

底色：#ecf3cf / 文字颜色：#ea5532

底色深、文字浅

如果和上面的例子相反，使用深的底色会有什么效果呢？下面的例子都在深的底色上使用了明度差异显著的文字颜色。

深的底色能为画面带来强烈的印象。通常情况下，这种配色不会用在大段正文中，往往用于需要靠颜色来突出的标题部分或商品包装上。

时光只解催人老，不信多情，长恨离亭，泪滴春衫酒易醒。
梧桐昨夜西风急，淡月胧明，好梦频惊，何处高楼雁一声。

底色：#004ea2 / 文字颜色：#7d6b48

在选择深的底色时要尽量避免上面那种因明度相近而造成阅读困难的情况，配色时要留意底色和文字颜色的对比。

时光只解催人老，不信多情，长恨离亭，泪滴春衫酒易醒。
梧桐昨夜西风急，淡月胧明，好梦频惊，何处高楼雁一声。

底色：#120605 / 文字颜色：#c9caca

时光只解催人老，不信多情，长恨离亭，泪滴春衫酒易醒。
梧桐昨夜西风急，淡月胧明，好梦频惊，何处高楼雁一声。

底色：#403632 / 文字颜色：#c9caca

时光只解催人老，不信多情，长恨离亭，泪滴春衫酒易醒。
梧桐昨夜西风急，淡月胧明，好梦频惊，何处高楼雁一声。

底色：#ed6c00 / 文字颜色：#ffffff

时光只解催人老，不信多情，长恨离亭，泪滴春衫酒易醒。
梧桐昨夜西风急，淡月胧明，好梦频惊，何处高楼雁一声。

底色：#00834b / 文字颜色：#ffffff

时光只解催人老，不信多情，长恨离亭，泪滴春衫酒易醒。
梧桐昨夜西风急，淡月胧明，好梦频惊，何处高楼雁一声。

底色：#004ea2 / 文字颜色：#bbd4ef

时光只解催人老，不信多情，长恨离亭，泪滴春衫酒易醒。
梧桐昨夜西风急，淡月胧明，好梦频惊，何处高楼雁一声。

底色：#5f67ae / 文字颜色：#fff200

第10章

排版样例

一般排版指对正文进行排版，因为一本书中包含多种资源，所以又可以细分为对多字体的应用、图片的处理、诗歌居左居右的排版等。每种情况又细分为不同的解决方式，下面我将会用具体的案例，教大家在遇到不同情况下的样式怎样书写。这分为从纸书中获取的排版样例、基于ePub技术的排版样例两种。

10.1 目录格式

书籍目录有「通栏排」和「分栏排」两种形式。工具书多采用分栏排的格式。

用条款氏（阶梯式）的排法，每级题与每级题之间，根据版心的情况，退一个或两格排版。目录内容较长的，可在语气处回行。回行的内容：如果标题是有序号的回行「齐字」排，没有序号的回行「缩格」排。

另外，目录内容最小的标题有两种版式，一种是按标题的级别，「缩格」排；一种是接排目录内容，距右版口「两个字」回行，回行「齐字」。

然后给标题添加对应的超链接就行了。

10.2 标题

标题，是文章的主题。层次的标题起着提纲挈领地揭示和概括该标题所属的内容，从而帮助读者理解内容并便于查阅的作用。根据它们的内容，标题又分为部、编、篇、章、节和小题。有多级标题的图书，目录中最少应收入两级。

标题的版式包括选择标题的字级、字体，还包括标题的位置、序号、字空、占行和转行等。为了醒目，突出标题，也可以在标题四周或标题字四周加排花边和在标题下加排底纹衬托。

标题的分级

在一本书中，有大小不同的各种标题，根据图书性质和开本大小安排字号字体，把一本书中最大的标题称为一级标题，然后按标题的层次排列为二、三级标题……依次类推。

标题分级在排版或校对时也便于掌握和统一标题的格式。

标题的字级、字体

标题的字级、字体依据图书类别、标题等级及用字原则来选择。

一般图书标题的字体以黑体、标宋体、宋体、楷体、仿宋体为主，不用隶书、行楷、魏碑或其他艺术字体。对马列、政法、哲史及科技类等较严肃的图书，不宜选择过于艺术化的字体；开本较小的图书或小屏设备，则不宜选择较大的字级；层次较多的标题用字，应遵循由大到小、由重到轻、变化有序、区别有秩的原则。图书标题选用的字级与正文主体文字所用字级的对比，不宜过于强烈。

在大小中见层次，在变化中显区别。切忌大小失序、轻重倒置。

期刊标题的字体、字级可根据期刊性质、文章内容、版面大小、篇幅长短、标题字数多少选择使用隶书、行楷、魏碑或圆头体、琥珀体等艺术字体，甚至还可将标题文字作左斜、右斜等变形处理，使版面更加生动

活泼。标题文字的字级可比正文主体文字大若干级，从而形成较强烈的对比，在版面上也显得更加突出。

图书标题序号的大小和字体应与标题相同。

作为标题使用的原则是字体相同，字号由大字号到小字号顺序排列；字体不同、字号相同的排列顺序是黑体、标宋、楷体和仿宋体。

标题位置

图书标题位置有居中、居左和居右三种样式。一、二级标题多居中；标题文字较多的，以居左为宜。

标题形式有单一型标题（一个主题）和复合型标题（主题前有引题或主题后有副题）两种。对于复合型标题，引题、主题、副题可用不同的字体、字级加以区别，且引题、副题的字级应小于主题。

标题字空

对字数较少的标题，为了不失衡，可适当增加字空。如果是在平板中阅读，可以使用媒体查询，单独为大屏设备设计一套样式。

标题占行

图书标题在版面上占一定的行数，既为字级大于正文的字级提供了必要的空间，同时又形成了标题在一定空间中鲜明、突出、醒目的效果。各级标题所占的行数，与标题等级相关。一级标题，可另页、另面，如果排正文，则最多可占一面行数的三分之一左右；二级标题，多占二至三行；三级标题，多占一至二行；四级以下标题，占一行或不占行。各级标题的占行数一经设定，全书必须统一。

两种特例：两级标题连排时，可省一行；某级标题沉底（即在该面最末一行），其他补救办法无法使用、这一面又有占行数较多的一级标题时，这级标题所占行数可减少一行。

期刊标题的占行主要根据版面的大小、标题字的多少和文章主体文字所占的面积等进行设计。尽量做到不要因标题占行过多而使正文主体文字显得压抑；也不要因标题占行太少而影响整个版面的均衡。

标题换行

图书标题长度超过版心宽四分之三时需换行；期刊标题文字行长超过正文行长五分之四时才换行。换行可顶格齐左、齐肩或居中三种形式。若上行居中，则换行也必须居中。转行时要按标题的语气转行，或在标点处转行（在标点处转行标点不排）；转行后的字数不低于5个字。不论何种换行方式，都必须严格遵守不损害文意的原则，即标题换行不能割

裂词义，虚词不能转为下一行的第一字等。

对标题层次设计时应注意：

1. 必须先了解全书或全套书的标题层次结构，全面考虑，尽量使整个标题的结构形式能表现出书籍内容的逻辑性。
2. 设计后的标题层次，应层次分明，前后统一，版式美观、风格一致，便于查阅。
3. 标题层次除上述所说情况外，还可运用梯形、数码、花边、线条、图案、网纹、色彩及各种排式等方法加以区别。对少儿读物、通俗读物、文学艺术类图书，标题设计应新颖生动、吸引读者。

作者栏的安排

图书作者的署名应排在封面上，可以在书名下居中横排，也可不居中；可紧挨着书名下，也可与书名有较大距离，要视封面整体的设计而定。如果书名竖排，则作者名也要竖排，这样比较协调、美观。作者名排在书名上方的也有，这种情况多是书名在封面的下方。

期刊文章的作者署名一般紧接着文章名称排，排列方式与文章名称排列方式一致。不论如何排列，一般作者栏的字号要小于书名（或文章名）的字号。编排的方式以整个封面（或版面）给人以美感为原则。

竖排标题版式

竖排标题分为有序号的书刊标题和无序号的期刊杂志标题。

有序号的竖排标题，每级标题与每级标题为阶梯式的版式，在标题距上版口的字距全书必须统一。如第一章距上版口两个字，第一节距上版口三个字，在排以后的章、节时均按这一字距排版。只要在排版中同级标题使用一个样式，它们的效果就是统一的。

无序号的竖排标题，分为横排文竖题和竖排文竖题。两种形式的竖排标题在占行的位置中偏上排。横排文竖题可以使用float属性排版。

下面以实际案例的方式，展示标题的排版。当然，这都是很简单的排版。

为了美化版面，标题设置了各种排列形式。标题的排列形式主要是根据标题的字数和语句的含义，自然形成标题的排列形式。

单行式。

第十章　□□□□□

□□□□□□□

单行的标题根据对齐方式可以有3种，居左、居中、居右。

标题置于左上顶部，并远离正文。标题与正文间距为30%，标题采用标题字体——小标宋，标题字号与正文字号相同。

标题依旧设置左上顶部，采用内文1.5倍字号的字体，这种方式的标题设计让标题和正文的对比尤为突出，同样标题与正文之间也存在一定的空间。设置成格式时，需要考虑一行容纳的字数，适合短标题。

标题位于正文的上方，字号是正文的1.35倍，标题与正文之间空一行。顶部至标题间距为30%。

标题与正文使用有对比的字体，比如标题为无衬线黑体，正文为衬线宋体。标题依旧位于正文的上方，字号是正文的1.25倍，标题与正文之间空一行。

给标题添加一条分割线，这种方式的好处是，将文本属性划分，并进一步强调了标题这一行的重要性。

当觉得版式枯燥无味时，可以尝试加入与内文匹配的视觉元素，比如圆形、图标、图案等。

均列式。

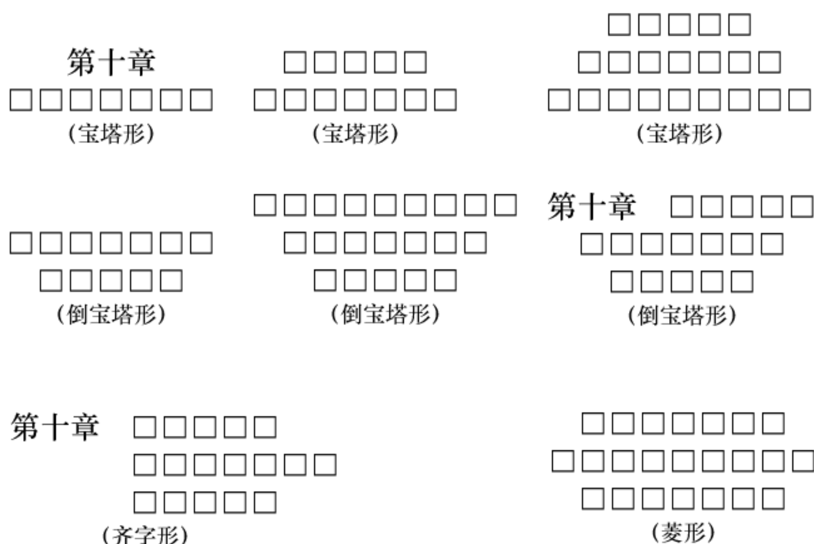


图10-2　标题的排版

对称式、斜列式。两种版式的要求是：第一行题字与左侧，最后一行题字与右侧，字距要相等。

对称式

□□□□□

　　　　　×××

□□□□□□

　　　　□□□□□□

斜列式

□□□□□□

　　□□□□□□

　　　　□□□□□□

低（缩）格式。低（缩）格式标题的版式要求是第二行标题字比第一行标题字低一格排，以此类推。

□□□□□□

　□□□□□□

　　□□□□□□

左齐式、右齐式。即左对齐和右对齐。

竖排标题样式。版式要求同「低（缩）格式」。有单行式、斜列式（梯形）、低格式（梯形）。

顶部出血。章节标题图加标题内容构成一张图，左中右贴边。这种效果的实现只需要设置盒子的外边距为负。

篇章页

如果简单的把标题粗排，显得不够美观。需要为一级标题设置类似封面一样的效果，灵感可以取自封面中的某个元素。篇章页设计好之后，有两种实现方法，一种是图片＋代码的方式实现，一种是完全用图片当作背景来达到目的。区别是在Kindle设备中，body元素不支持背景图的显示。当你在制作电子书时，请先考虑好适用的硬件设备，或者在电子书文件中提示该版本那些软件或硬件可以完整的显示你预先设置的效果。

10.3　正文

正文版式是图书内容最基本的版式。各类图书都可以按照正文版式的要求排版

一般正文需要设置以下样式：字体系列、字号、行高、首行缩进、段落间距、对齐方式等。除了正文字体外还需要些其他字体的正文样式，比如黑体段落、楷体段落、仿宋段落等。

正文文字的排法

密排。密排是常规的字距，按照文字字号的幅面顺序排版，文字与文字之间没有字距。密排是图书内容最基本的排法。

疏排。疏排是美化版面的字距，在文字与文字之间，有空格的距离（字距）。一般用于小学教材、儿童读物、奖状、说明书和请柬等印刷物。CSS中使用letter-spacing属性控制字距。

10.3.1　一般正文

下面是正文文字排版最基本的版式。起行低两格（两个字的宽度），回行顶格（回行齐口，版心的左版口，默认版口）。

常规版式

下面以一个实际的案例来讲解下各个属性怎样设置。下面是需要排版的正文：

　　近日，解放军报发表文章《谨防王者荣耀「粘手」》，报道认为目前军营中部分官兵沉迷「王者荣耀」游戏，对官兵健康、训练带来不良影响，警示这种现象将危害部队战斗力。

　　报道指出，自从部队放开智能手机使用后，官兵可以充分享受移动互联网的利好，但不少官兵却迷上「王者荣耀」。有些基层带兵人反

映，只要一到周末休息，有的班推门看去，几乎全班战士手里捧着手机，全神贯注盯着屏幕：「敌军在5秒钟后到达」「全军出击」的游戏声不绝于耳。

HTML代码：

```
<p class="bodycontent">近日，解放军报发表……部队战斗力。</p>
<p class="bodycontent">报道指出，自从部……声不绝于耳。</p>
```

CSS代码：

```
p.bodycontent {
    margin: .5em 0; /*段落间距*/
    font-family: "宋体"; /*标准的正文字体*/
    font-size: 1em; /*正文字体大小，默认是16px*/
    line-height: 1.5em; /*行高1.5em，标准正文字的1.5倍*/
    text-indent: 2em; /*首行缩进2个汉字空间*/
    text-align: justify; /*两端对齐*/
    text-justify: inter-ideograph; /*用表意文本来排齐内容*/
    color: #000; /*正文字体颜色*/
    word-break: break-all; /*允许在单词内换行*/
}
```

对于text-justify属性，中文书用inter-ideograph值最合适。关于引用的外部字体——书宋。CSS代码：

```
@font-face {
  font-family: "宋体";

src: local("st"),local("st"),local("SongTi"),local("Songti"),
       local("正文"),local("宋体"),local("明体"),local("明朝"),
       local("Songti"),local("Songti SC"),local("Songti TC"),
       local("STSongti"),local("STSong"),local("Song S"),
       local("STBShusong"),local("TBMincho"),
       local("HYMyeongJo"),local("DK-SONGTI"),
  url("../Fonts/YaSong.ttf");
}
```

设置了`word-break`之后，不在一行的英文单词，在阅读器中会自动添加连字符。

还有一种情况是段与段之间有一个空行区分了，这种可以用`<p><br/></p>`或者单独书写一个样式，设置`margin-top`。如：

报道指出，自从部队放开智能手机使用后，官兵可以充分享受移动互联网的利好，但不少官兵却迷上「王者荣耀」。有些基层带兵人反映，只要一到周末休息，有的班推门看去，几乎全班战士手里捧着手机，全神贯注盯着屏幕：「敌军在5秒钟后到达」「全军出击」的游戏声不绝于耳。

报道警示称，网络游戏是一把双刃剑。偶尔玩一会儿，对于整天忙碌的战士来说不啻于枯燥生活的调味剂、烦恼压力的倾泻品。但是不得不承认，过度沉溺于手机游戏，正在悄悄损害官兵的身心健康，甚至给安全管理、战斗力带来危害。长时间玩网络游戏，容易导致身体免疫力下降、内分泌失调，情绪易燥、易怒，容易患神经衰弱症；玩游戏上瘾后，就会减少与周边人交流，长此以往容易形成孤僻性格，甚至患上自闭症；沉溺于网络游戏，人的精神高度集中，伴随着血液循环加速、心跳加快，体力、精力消耗很大，容易过度疲劳。最近，人民网在四天内从不同方面三评「王者荣耀」的危害性，说明它已经引起社会广泛关注。

HTML代码：

```
<p class="bodycontent">报道指出，自从部……声不绝于耳。</p>
<p class="bodycontent-top">报道警示称，网络游戏……引起社会广泛关注。</p>
```

CSS代码：

```
p.bodycontent-top {
    margin: 2em 0 .5em; /*段落间距上为2em（根据块元素的特性有0.5em和上一段合并在一起了，所以为了空出一行，需要设置为2em），左右边距为0，下外边距为0.5em*/
    font-family: "宋体"; /*标准的正文字体*/
    font-size: 1em; /*正文字体大小，默认是16px*/
    line-height: 1.5em; /*行高1.5em，标准正文字的1.5倍*/
    text-indent: 2em; /*首行缩进2个汉字空间*/
    text-align: justify; /*两端对齐*/
    text-justify: inter-ideograph; /*用表意文本来排齐内容*/
    color: #000; /*正文字体颜色*/
    word-break: break-all; /*允许在单词内换行*/
}
```

然后是其他字体的正文，这里就不再书写了，无非是某个属性的更改。

条款和剧本版式

用于条款类的图书或剧本等印刷物的排版。起行顶格回行齐肩（齐字）或者约定缩格字数。条款版式根据其版式的形式又称为「阶梯形版式」。特点是条理清楚，因其序号在行首，子目依次缩格便于查找。

第一条　为了保护劳动者的合法权益，调整劳动关系，建立和维护适应社会主义市场经济的劳动制度，促进经济发展和社会进步，根据宪法，制定本法。

第十一条　地方各级人民政府应当采取措施，发展多种类型的职业介绍机构，提供就业服务。

第二十一条　劳动合同可以约定试用期。试用期最长不得超过六个月。

第一百零一条　用人单位无理阻挠劳动行政部门、有关部门及其工作人员行使监督检查权，打击报复举报人员的，由劳动行政部门或者有关部门处以罚款；构成犯罪的，对责任人员依法追究刑事责任。

上面的案例首行缩了3个汉字，为了对齐最长的「第一百零一条」，我给前面几条添加了字距，但这可能不规范。这里只是提供一个使用CSS的思路。

HTML代码：

```
<p class="clause">第<span class="one-character">一</span>条　……</p>
<p class="clause">第<span class="two-character">十一</span>条　……</p>
<p class="clause">第<span class="three-character">二十一</span>条　……</p>
<p class="clause">第一百零一条　……</p>
```

CSS代码：

```
/*条款*/
p.clause {
    margin: 0.5em 0 0.5em 3em;
    font-family: "宋体";
    font-size: 1em;
    line-height: 1.5em;
    text-indent: -3em;
    duokan-text-indent: -3em;
    text-align: justify;
```

```
    text-justify: inter-ideograph;
    color: #000;
    word-break: break-all;
}
/*一个序号字符*/
span.one-character {
    border-left: 1.5em solid transparent;
    border-right: 1.5em solid transparent;
}
/*二个序号字符*/
span.two-character {
    border-left: 1em solid transparent;
    border-right: 1em solid transparent;
}
/*三个序号字符*/
span.three-character {
    border-left: 0.5em solid transparent;
    border-right: 0.5em solid transparent;
}
```

一开始是想要用margin属性来控制序号字符两边的距离，但是在多看阅读器中没有效果，所以换成用border来实现，只要把边框变成透明就行了。

（幕启：北风呼啸，四壁昏暗；李奶奶捻灯，屋中转明。）

李奶奶　唱【西皮散板】

打渔的人经得起狂风巨浪，

打猎的人【原板】哪怕虎豹豺狼。

看你昏天黑地能多久！

革命的火焰一定要大放光芒。

（铁梅挎货篮进屋。

铁　梅　奶奶！

李奶奶　铁梅！

HTML代码：

```
<p class="scene">（幕启：北风呼啸，四壁昏暗；李奶奶捻灯，屋中转明。）</p>
<p class="play">李奶奶　唱【西皮散板】</p>
<p class="dialogue">打渔的人经得起狂风巨浪，</p>
<p class="dialogue">打猎的人【原板】哪怕虎豹豺狼。</p>
```

```
<p class="dialogue">看你昏天黑地能多久！</p>
<p class="dialogue">革命的火焰一定要大放光芒。</p>
<p class="scene">（铁梅挎货篮进屋。</p>
<p class="play">铁　梅　奶奶！</p>
<p class="play">李奶奶　铁梅！</p>
```

CSS代码：

```
/*剧本*/
p.play {
    margin: 0.35em 0;
    font-family: "宋体";
    font-size: 1em;
    line-height: 1.5em;
    text-indent: 0em;
    duokan-text-indent: 0em;
    text-align: justify;
    text-justify: inter-ideograph;
    color: #000;
    word-break: break-all;
}
/*台词*/
p.dialogue {
    margin: 0.35em 0;
    font-family: "宋体";
    font-size: 1em;
    line-height: 1.5em;
    text-indent: 6em;
    duokan-text-indent: 6em;
    text-align: justify;
    text-justify: inter-ideograph;
    color: #000;
    word-break: break-all;
}
/*场景*/
p.scene {
    margin: 0.5em 0 0.5em 3em;
    font-family: "宋体";
    font-size: 1em;
    line-height: 1.5em;
    text-indent: 4em;
    duokan-text-indent: 4em;
    text-align: justify;
    text-justify: inter-ideograph;
    color: #000;
    word-break: break-all;
}
```

一般图书正文中可能会穿插一些别的书籍内容，这种情况我们只要参考下文中宋词排版即可。比如引文段落一般左外边距是有2个全角空格大的间距，引文上下与正文有一定的距离。部分引文紧接下面的正文是不首行缩进的，即无缩进段落。这个在排版中一定参考PDF，确认无缩进是不是存在。如果存在，就需要保留。

安徽籍工部左侍郎吕贤基上奏说：

> 今日事势，譬之于病，元气血脉，枯竭已甚，外邪又炽，若再讳疾忌医，愈难为救。为今之计，不妨效仿祖宗办法，着令各省官绅自办团练，则国体可报，匪患亦不难消矣。

此疏一上，咸丰帝茅塞顿开，马上采纳。大清国于是遍地办起团练。

咸丰三年（公元1853年）二月，太平军攻破安徽省城安庆，巡抚蒋文庆战殁，帮办团练大臣周天爵败走。太平军一鼓作气连占含山、巢县、和州之裕溪口。

10.3.2　书信

书信，又称书、启、笺、牍、简、札等。是人们以书写于纸张（或其他的文字记录体）的文字、图像为内容的一种交流形式。

书信的主要结构有前文、正文、后文。前文一般就是称谓，正文就是书信的主要内容，后文一般是祝福语、日期、发件人等。

西谛先生：

昨乔峰交到惠赠之《中国文学史》三本，谢谢。

去年冬季回北平，在留黎厂得了一点笺纸，觉得画家与刻印之法，已比《文美斋笺谱》时代更佳。譬如陈师曾齐白石所作诸笺，其刻印法已在日本木刻专家之上，但此事恐不久也将销沉了。

因思倘有人自备佳纸，向各纸铺择优各印数十至一百幅，纸为书页形，采色亦须更加浓厚，上加序目，订成一书，或先约同人，或成后售之好事，实不独为文房清玩，亦中国木刻史上之一大纪念耳。

不知先生有意于此否？因在地域上，实为最便。且孙伯恒先生当能相助也。

此布，并颂

曼福

迅　启上

二月五日

上面书信一共有5个样式，分别是：称谓、正文、祝福语、署名、日期。它们的样式都是基于正文样式的，我只修改了某些声明，比如对齐方式、行高、外边距等。故这里不再贴出CSS代码，你可以使用Sigil打开这个ePub文件，查看具体代码。

致許家惺 一八九八年二月十七日

默齋仁兄先生大人足下：

　　别後次晨到硖，乘王升记翰船，午刻開行，晚抵平湖，次日巳刻始逵上海，謁兄穰卿、颂閣兩先生。途中平善，堪慰垂注，辰维起居佳暢爲颂。弟在此同得徒褚君子後，與聞绪输，甚幸甚幸。足下爲我導夫先路，感何可言。雲樵先生人極和平，惟言語不通，無從請益。愷君、敬堂二先生亦待弟甚周到，堪告慰耳。前日穰卿先生與卓如先生信，擬請鄭蘇盦先生爲正主筆，此刻尚未定，因外人颇不満於穰卿先生著述故也。第二期報出，粤人颇有議之者。瑣此奉告。即請

撰安不宣

弟王国维顿首　正月廿七日

这个属性内容比上面的书信案例，多了标题和标题中的日期两个样式。

HTML代码：

```
<p class="letter-title">致許家惺 <span class="letter-data">一八九八年二月十七日</span></p>
```

CSS代码：

```
/*书信标题*/
p.letter-title {
    margin: 1em 0; /*段落间距*/
    font-family: "小标宋"; /*标题字体*/
    font-size: 1.15em; /*正文字体1.15倍大小*/
    line-height: 1em; /*行高1em*/
    text-indent: 0em; /*首行缩进0个汉字空间*/
    text-align: center; /*居中对齐*/
    color: #000; /*正文字体颜色*/
```

```
}
/*书信标题中的日期*/
span.letter-data {
    font-family: "宋体"; /*标准的正文字体*/
    font-size: .75em; /*父级字体大小的0.75倍*/
}
```

然后对于书信的整体概念而言，推荐给书信内容添加一个div，用于包裹书信内容。并给这个div设置上下外边距，以与正文有一定间距存在。因为上述演示的原因，div没有添加。你可以在实际的排版中添加进去。

10.3.3 诗词曲

古体诗，是中国古代的一种文学体裁，与乐府诗并称为「汉代文学双璧」或「双葩」。古体诗简称古诗，是与「近体诗」相对，故称。

唐朝以前的诗歌，不分古体近体。初唐开始，律诗和绝句的格式固定下来，以前那些形式格律不严格不讲究的诗体，就一律称作古体，并把律诗绝句合称为近体。即使在唐代抑或以后，仍有许多诗人依古体写作诗篇。「古体」云云并不含有时代分期的意思。

形式特点上，古体近体的差别在于，古体诗是一种较少拘束的诗体，句子数目不拘，篇幅不限，每句字数不拘，不严格讲究平仄与对仗，也不讲究押韵方式，可以兼用平声韵与仄声韵。

古体诗最基本的形式是五言古诗，以及七言古诗。全部体裁则可分为七种：四言、五言、七言、五七杂言、三七杂言、三五七杂言、错综杂言。

近体诗，又称格律诗，是中国讲究平仄、对仗和押韵的诗体。

今体诗，唐代形成的格律诗体。也因其字数、句数、平仄和用韵等都有一定的严密格式和规律，犹如法律、纪律一样，所以后人也称其为律体诗。

综上，基于排版的需求，对诗歌的排版主要有两种：居中、居左。格式工整、字数对称的一般采用居中，但如果每句字数过多也会采用居左的方式排版。

古体诗

今日中国最广为人知的古体诗多为五言诗或七言诗，具代表性的有《古诗十九首》、陈子昂之《登幽州台歌》等。本小节节选二首进行样例的排版。

之十五

生年不满百，常怀千岁忧。

昼短苦夜长，何不秉烛游。

为乐当及时，何能待来兹。

愚者爱惜费，但为后世嗤。

仙人王子乔，难可与等期。

——《昭明文选·卷二十九》

也可以这么排版：

继北方《诗经》以后，南方《楚辞》兴起，我国最早之伟大诗人屈原，即受南方民歌影响写出《离骚》、《九歌》、《九章》等名著。由于楚物、楚语、楚声与北方不同，一般不以四字句为主，或用五字，或用七字，或用对仗，且每句中具有「兮」字。如《云中君》云「石濑兮浅浅，飞龙兮翩翩」，则为五字句。如《山鬼》云「雷填填兮雨冥冥，猿啾啾兮狖夜鸣。风飒飒兮木萧萧，思公子兮徒离忧」，则为七字句，每句两用叠字，此亦为《诗经》所无。至《悲回风》八句，连用六叠字；宋玉《九辨》十二句，连用十一叠字。此在后来辞赋中亦系少见。

汉魏六朝以来，五言诗盛行，《古诗十九首》中有一首云：

> 青青河畔草，郁郁园中柳。盈盈楼上女，皎皎当窗牖。娥娥红粉装，纤纤出素手。

用六句五言诗，每句首二字皆叠字，顾炎武《日知录》卷二十一，以为此诗「连用六叠字，亦极自然，下此即无人可继」，可见赞扬之至。

诗歌采用的是这个样式。HTML代码：

```
<div class="poetry-quote">
    <p>青青河畔草，郁郁园中柳。盈盈楼上女，皎皎当窗牖。娥娥红粉装，纤纤出素手。</p>
</div>
```

涉及的CSS：

```
/*诗词引文*/
div.poetry-quote {
    margin: .75em 0 .75em 2em;  /*给诗词盒子做外边距加2个字符间距
```

```
*/
}
div.poetry-quote p {
    margin: .35em 0;
    font-family: "仿宋";
    font-size: 1em; /*正文字体大小，默认是16px*/
    line-height: 1.35em; /*行高1.35em，标准正文字的1.35倍*/
    text-indent: 2em; /*首行缩进2个汉字空间*/
    text-align: justify; /*两端对齐*/
    text-justify: inter-ideograph; /*用表意文本来排齐内容。*/
}
```

字数不等的古体诗：

登幽州台歌

陈子昂

前不见古人，后不见来者。

念天地之悠悠，

独怆然而涕下。

字数不等且多段的杂言古诗：

梦游天姥吟留别

李白

海客谈瀛洲，烟涛微茫信难求。

越人语天姥，云霞明灭或可睹。

天姥连天向天横，势拔五岳掩赤城。

天台一万八千丈，对此欲倒东南倾。

我欲因之梦吴越，一夜飞度镜湖月。

湖月照我影，送我至剡溪。

谢公宿处今尚在，渌水荡漾清猿啼。

脚著谢公屐，身登青云梯。

半壁见海日，空中闻天鸡。

千岩万转路不定，迷花倚石忽已暝。

熊咆龙吟殷岩泉，栗深林兮惊层巅。

云青青兮欲雨，水澹澹兮生烟。

列缺霹雳，丘峦崩摧。

洞天石扉，訇然中开。

青冥浩荡不见底，日月照耀金银台。

霓为衣兮风为马，云之君兮纷纷而来下。

虎鼓瑟兮鸾回车，仙之人兮列如麻。

忽魂悸以魄动，恍惊起而长嗟。

惟觉时之枕席，失向来之烟霞。

世间行乐亦如此，古来万事东流水。

别君去兮何时还？

且放白鹿青崖间，须行即骑访名山。

安能摧眉折腰事权贵，使我不得开心颜！

在上面的例子中需要给分段的第一句添加上外边距`margin-top`。HTML代码：

```
...
<p class="poetry-top">海客谈瀛洲，烟涛微茫信难求。</p>
...
<p class="poetry-top">我欲因之梦吴越，一夜飞度镜湖月。</p>
...
<p class="poetry-top">世间行乐亦如此，古来万事东流水。</p>
...
```

CSS代码：

```
/*诗歌分段上空*/
p.poetry-top {
    margin-top: 0.875em!important;
}
```

近体诗

近体诗的体裁有五言绝句、七言绝句、五言律诗、七言律诗、六言律句等。

五言绝句：

鸟鸣涧 王维

人闲桂花落，夜静春山空。

月出惊山鸟，时鸣春涧中。

至于在注释中有注释图片的情况我已经在9.2节中说过，可参考。

七言律诗：

询诸当代大数学家陈省身，大物理学家杨振宁论科学家心目中或微观世界之美，皆简洁合度，恰到好处。诚如天体物理学家开普勒所云，宇宙是六声部的大交响。其中有数、有律，这就是秩序，而秩序则必以和谐为其前提。在混沌中放出光明，乃是科学巨人和大艺术家之本分。我曾在所描绘陈省身与杨振宁先生一画「奇文共欣赏，疑义相与析」中题七律一首：

纷繁造化赋玄黄，宇宙浑茫即大荒。
递变时空皆有数，迁流物类总成场。
天衣剪掇丛无缝，太极平衡律是纲。
巨擘从来诗作魄，真情妙悟铸文章。

杨振宁先生激赏此诗，以为方之英国大诗人蒲柏（Alexander Pope）或无多让。今以谈及辛稼轩晚年之诗《永遇乐》与《南乡子》，可以看到凡至极境，古今中外、科学、文艺必有相通之处。亦有以告于当今之从事者，万勿侥幸以求捷径。终南之下有卢藏用而北山有周颙者，皆以其伪而为世所不屑，能不自警欤？

诗歌正文单独一个div，然后给这个div元素设置上下外边距；div元素里的诗歌正文设置样式。

宋词

词开始出现，是在中国的唐代。更精确一点，是在中国初唐时期。伴随着当时「胡乐」传入，「燕乐」大盛，词也逐渐脱离传统的五言古诗、七言古诗，成为一门独立的诗歌艺术。

按长短规模来分，词大致可分为小令（58字以内）、中调（59～90字）和长调（91字或以上）。一首词，有的分为一段，称为单调；有的分为两段，称为双调；有的分为三段或四段，称为三叠或四叠。

同一年辛稼轩再一次作《南乡子·登京口北固亭有怀》一词，怀古之情依旧，而豪情更有胜于《永遇乐》：

何处望神州。满眼风光北固楼。千古兴亡多少事，悠悠，不尽长江滚滚流。

年少万兜鍪，坐断江南战未休。天下英雄谁敌手？曹刘。生子

当如孙仲谋。

辛稼轩晚年之词老辣跌宕，气宇高迈，显为斫轮老手。信笔挥洒而不失周赡；激越之中蕴有冲融；刘勰所谓：「纷哉万象，劳矣千想。元神宜宝，素气资养。水停以鉴，火静而朗。无扰文虑，郁此精爽」。（《文心雕龙·养气》）这「精爽」二字，谈何容易？

这种是两阕分段了，而有的词两阕未分段，在阕与阕之间使用两个全角空格分开。如下：

从己之忆人，推到人之忆己，又从相忆之深，推到相见之难。文字全用赋体白描，不着粉泽，而沉哀入骨，宛转动人。南唐二主之尚赋体，当受韦氏之影响。又如《荷叶杯》云：

记得那年花下。深夜。初识谢娘时。水堂西面画帘垂。携手暗相期。　　惆怅晓莺残月。相别。从此隔音尘。如今俱是异乡人。相见更无因。

此词伤今怀昔，亦是纯用白描。自「记得」以下直至「相别」，皆回忆当年之事。当年之时间，当年之地点，当年之情景，皆叙得历历分明，如在昨日。「从此」三句，陡转相见无因之恨，沉着已极。又如《女冠子》两首，写梦中之情景，亦真切生动。词云：

四月十七。正是去年今日。别君时。忍泪佯低面，含羞半敛眉。　　不知魂已断，空有梦相随。除却天边月，没人知。

昨夜夜半。枕上分明梦见。语多时。依旧桃花面，频低柳叶眉。　　半羞还半喜，欲去又依依。觉来知是梦，不胜悲。

前首记去年离别之情，后首记梦中相遇之情，皆刻画细微，如见其面，如闻其声。两结句重笔翻腾，畅发尽致，尤觉哀思洋溢，警动无比。蜀自李白以还，若韦氏者，可谓第二大词人矣。

上面举的例子是从一本书中截取的，故头尾两端没有首行缩进。

元曲的排版与诗词类似，这里就不再讲了，可参考纸书书写排版样式。

10.3.4　古体诗歌自适应换行

本篇文章是 @云轩阁主 写的，这里录入一下，文章原名制作自适应换行的排版效果。部分代码有调整。

前言

这个标题大家看着可能摸不着头脑，先简单普及几个定义。

在HTML中，文字换行有两种类型，一个是块级标签定义段落。比如说第一段到这一段之间的换行，就是两个`<p>`标签（或者两个`<div>`标签）。简而言之，一个`<p>`到`</p>`之间就是一个完整的自然段。段与段之间的距离，就是段间距，可以通过定义`<p>`或者`<div>`的margin属性来控制。

另外一种就是`<br/>`，这个实际上是一个行内标签（in-line），也就是只能在一个段落内部起作用的标签，不能跨段落使用。这个标签的作用是行内之间换行，但是换行后的两段文字还是被视为一段文字的，所以这两段文字之间的间距只是行间距，也就是一个自然段内上下两行文字的距离。

不同设备的自动换行

对于阅读器而言，它会根据设备的物理宽度来自动给文字换行，相当于程序自动在设备边缘给文字加了个`<br/>`标签。

不同的设备，因为宽高比和分辨率不一样，自动换行的长度也不一样，下面是一个对比图。

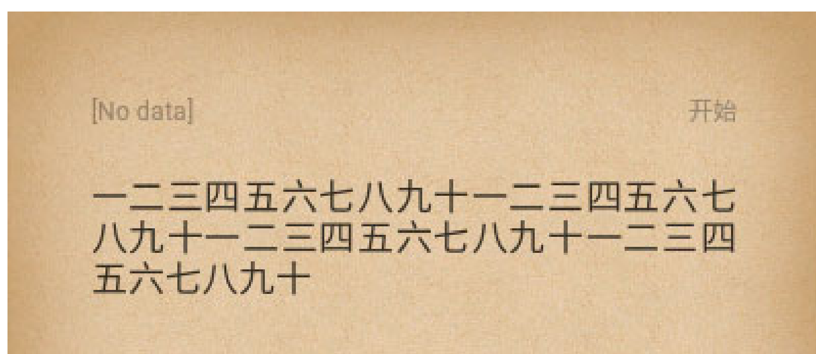


图10-4　手机显示界面

左图是宽高比为4:3的平板上多看的自动换行效果，大概是二十三个字符左右。当然如果你手动放大文字字号，字符数也会相应减少。右图是宽高比为16:9的手机上多看的自动换行效果，可以看到手机的长度更长，但相应的宽度只有十七个字符就要自动换行。

诗歌的自动换行实例

举个实际的例子。如下面左图所示：

[No data]

《出塞》·王昌龄

秦时明月汉时关，万里长征
人未还。

但使龙城飞将在，不教胡马
度阴山。

图10-6　手机显示七言诗

这个是平板上显示诗歌的排版，对于七言绝句而言，一行显示两句，总共显示两行是典型的排版（左图）。但是同样的排版到了手机上就是右图的样子了。

原因就是上面所说，手机宽度较小，所以产生了自动换行。当然我们可以用一个折中的办法，就是诗歌固定一句一行，如下图：

[No data]

《出塞》·王昌龄

秦时明月汉时关，

万里长征人未还。

但使龙城飞将在，

不教胡马度阴山。

图10-7　七言诗一行一句

这样一来平板和手机默认字体大小显示的就都不会自动换行了（当然手动放大字号肯定会自动换行，这个没法避免，我们只能保证默认字体的自适应），但是我们可以比较一下，这样一来牺牲的是平板设备两侧的空余空间，如果字体较小的话，排版还是显得不够均匀。

自适应换行排版

我们的目标是在平板上能一行显示两句，而在手机上又能自动切换为一行一句。简而言之，就是不要让程序在设备边缘自动换行，而让它在我们想要的地方自动换行。

实际操作的代码如下：

```
<p class="title">《出塞》·王昌龄</p>
<div class="pad-phone-poetry">
  <p><span>秦时明月汉时关，</span><span>万里长征人未还。</span></p>
  <p><span>但使龙城飞将在，</span><span>不教胡马度阴山。</span></p>
</div>
```

CSS代码：

```
p.title {
    text-align: center;
}
div.pad-phone-poetry p {
    margin: .5em 0;
    font-family: "楷体";
    text-align: center;
    white-space: nowrap;
}
```

平板的显示效果还是同第三张图，而放到手机上显示效果如下：

[No data]

《出塞》·王昌龄

秦时明月汉时关，
万里长征人未还。

但使龙城飞将在，
不教胡马度阴山。

图10-8　七言诗一行一句

我们看一下代码，实际上要关注的只有两个地方，第一是诗歌的属性中，需要有`white-space:  nowrap;`这个属性，不允许自动换行；第二是在诗歌本身段落中，在需要成为整句的地方用行内标签括起来（`<span>`或者`<a>`都可以），这样一来，在手机上程序就会优先从`<span>`之间进行自动换行了。

可以看到前两句和后两句之间的间距较大，这个原因在前言里已经说明了，第一二句之间实际上是`<br/>`标签，而二三句之间是`<p>`标签，二者的行距定义不同。

要解决这个问题，可以在诗歌属性中定义line-height，这控制了`<br/>`标签的行距，也可以去掉诗歌段落中的`<p>`标签。但要注意，如果平板宽度较大，可能出现第一行三句，第二行一句的情况。

总结

本篇的自适应换行技巧当然也有缺点，它比较适合用于古典诗歌文学，尤其是每句字数固定的，否则就可能出现每一行换行不一致的情况。

当然这个技巧不光可以用于诗歌，也可以用于章节标题等特定设计，因为其本质是让段落在我们想要的地方自动换行。会不会玩崩还是要看排版的调整。

10.4　插图

插图是插在正文内容之中的图片，它是正文叙述内容的形象说明，使正文形象化、立体化。所以，插图的排版位置和排版规范尤为重要。

插图排版位置的原则

书刊插图的排版原则是先见文，后见图，图随文走的原则（插图不能超过小题，原则上不能超过居中排的节题）。

先见文指的是（见图×），在版心允许的情况下最好接排在（见图×）自然段结束的正文下面。插图必须按照顺序排列。

插图的串文

插图的大小不等，而插图在版心中要求疏密得当，较小的插图就需要图旁串文。表现出来，即绕排。

插图串文规范是插图不超过版心宽的2/3。

插图与正文之间的距离，要大于正文行距的距离，应在一字左右。

需要串文的插图如果在一面上只有一个图时，原则上应按「先外后内（先右后左）」的顺序确定其位置。如果有两个插图时应对角交叉排，上图排在右版口一边，下图应该排在左版口一边。如果版式许可也可以将两张图并列居中排。

插图说明的格式

插图说明包括图题和图注两部分，术语称为「图文」。其字体、字号可参考原书，一般都小于正文字号。

①串文插图。图说明左右不超过插图宽度。

②通栏插图。左右缩进版心个3个字。

③图说明可以接排或分栏排（近似无线表），并列图。

④图题或图注内容。居中或居左排。

图书中的插图分为三个结构：图片标题、图片、图片说明三种。我们需要把这三个内容放在一个盒子（`<div>`）里，这个盒子可以设置宽度为100%，即屏幕宽度。然后分别设置图片的宽度（100%、80%、60%……），图片标题一般只需要设置居左和居中样式，图片说明设置居左、居中、居右三种。为了区别于正文，可以适当更换字体系列和字号，也可以加颜色。

插图的种类有单张、两张并列、三张及以上的画廊、行内生僻字、公式等几种。对插图的处理有的可以参考纸书排版，有的可以基于HTML的优势进行扩展。

10.4.1 单张插图

插图是图书的重要组成部分，正确处理插图在文中的排版非常重要。

下面是单张插图的一个例子。

中国方面损失军舰四艘，其中「致远」因炮弹用尽，管带邓世昌无奈之下，决定用船体去撞日舰时，被日舰所发鱼雷命中，全船官兵无一幸存。



北洋海军「致远号」官兵

战后，日舰队先撤，北洋舰队奉丁汝昌之命急驶旅顺港躲避。

HTML代码：

```
<p class="bodycontent">中国方面损失军舰四艘，其中「致远」因炮……官兵无一幸存。</p>
<div class="duokan-image-single">
   <img class="picture-80" alt="北洋海军「致远号」官兵" src="../Images/zhiyuanhao.jpg"/>
   <p class="duokan-image-maintitle">北洋海军「致远号」官兵</p>
</div>
<p class="bodycontent">战后，日舰队先撤，北洋舰队奉丁汝昌之命急驶旅顺港躲避。</p>
```

CSS代码：

```
div.duokan-image-single {
    margin: .5em auto; /*插图盒子上下外边距为0.5em，左右设置auto是为了水平居中这个盒子*/
}
img.picture-80 {
    margin: 0; /*清除img元素的外边距*/
    width: 80%; /*预览窗口的宽度*/
    box-shadow: 3px 3px 10px #bfbfbf; /*给图片添加阴影效果*/
}
p.duokan-image-maintitle {
    margin: 1em 0 0; /*图片说明的段间距*/
    font-family: "楷体"; /*图片说明使用的字体*/
    font-size: .9em; /*字体大小*/
    text-indent: 0; /*首行缩进为零，当你使用单标签p来指定首行缩进为2em时，记得在需要居中的文本中清除缩进，因为样式是叠加的*/
    text-align: center; /*图片说明水平居中*/
    color: #a52a2a; /*字体颜色*/
    line-height: 1.25em; /*行高，防止有很长的图片说明*/
}
```

duokan-image-single和duokan-image-maintitle是多看官方提供的API，样式命名成这种，会在多看阅读器中点击图片可以放大，同时页面底部显示图片说明。

关于引用的外部字体——楷体。CSS代码：

```
@font-face {
  font-family: "楷体";
  src: local("kt"),local("KaiTi"),local("STKaiti"),local("STKai"),
       local("楷体"),local("楷体_GB18030"),local("楷体_GB2312"),
       local("Kaiti"),local("Kaiti SC"),local("Kaiti TC"),
       local("MKai PRC"),local("MKaiGB18030C-Medium"),
       local("MKaiGB18030C-Bold"),local("DK-KAITI"),
  url("../Fonts/KaiTi.ttf");
}
```

10.4.2 交互图

交互图指的是读者可以用单击或双击的方式进入一个操作模式，执行比如放大、缩小、拖动等操作。

多看阅读器中图片交互

在上一节中，我们使用了多看官方提供的样式名，在多看阅读器中可以触发图片交互效果。这个是属于多看开放计划中一个效果。交互图完整代码如下：

```
<div class="duokan-image-single">
  <img src="../Images/tree.png" alt=""/>
  <p class="duokan-image-maintitle">主标题：大自然</p>
  <p class="duokan-image-subtitle">副标题：森林中的树</p>
</div>
```

为了保证点击放大之后的图像呈现效果，采用交互模式的图像数据应该保证足够的分辨率。

注意：

1. 主标题和副标题可以不出现；
2. 主标题和副标题可以在img之前出现；
3. 交互图不可以嵌套出现。

只要样式名命名成官方提供的样式名即可，至于如何显示图片、图片说明，制作者可以自己写CSS。下图在多看阅读中无交互效果：



图10-9　多看阅读器下非交互图

HTML代码：

```
<div class="picture_box">
  <img class="picture-100" alt="电子书设备" src="../Images/
ebook.jpg"/>
</div>
```

CSS代码：

```
div.picture_box {
    width: 100%;
    text-align: center;
    padding: .5em 0;
}
div.picture_box img {
    box-shadow: 3px 3px 8px #bfbfbf;
}
img.picture-100 {
    width: 100%;
}
```

只要不是用阅读器提供的样式名，在阅读器中就不会有交互效果。不过这也要看阅读器对图片的默认处理方式，有的阅读器，对ePub中的所有图片都默认交互。

掌阅iReader阅读器中图片交互

在掌阅iReader阅读器中，如果对图片不进行设置，默认图片都是可以交互的。它的代码与多看版不同，需要在`<img>`标签中添加扩展属性。然后使用解压软件打开ePub文件，在META-INF文件夹里添加一个扩展名为`.xml`的文件。此XML文件命名为「zhangyue-expansion.xml」（使用记事本新建一个文本文档，粘贴下面的代码，然后重命名文件名并修改文件扩展名），内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<zhangyue-expansion version="3.4.2">
  <book_id></book_id>
</zhangyue-expansion>
```

这个XML文件是必须要有的。一定要用解压软件打开ePub文件添加！

如上图如果想要在阅读器中没有交互效果，则HTML代码如下：

```
<div class="picture_box">
  <img class="picture-100" alt="电子书设备" zy-enlarge-src="none" src="../Images/ebook.jpg"/>
</div>
```

下图在掌阅iReader阅读器中没有交互效果：



图10-10　掌阅阅读器下非交互图

掌阅交互图，可以按照如下代码添加相应代码：

```
<div class="picture_box">
  <img class="picture-100" alt="电子书设备" zy-enlarge-src="self" src="../Images/ebook.jpg"/>
</div>
```

下图在掌阅iReader阅读器中有交互效果：



图10-11　掌阅阅读器下交互图

交互图在iOS端会有阴影，在安卓端会有白色边框。

另外在Lithium阅读器（点击图片）、Gitden阅读器（双击图片）中，图片默认交互。

10.4.3　并列图

并列图指的是两张尺寸一致的图片横向并列排放。并列图的排版主要是采用table元素来放置图片和图片说明。案例如下：



李鸿章1895年3月远赴东洋与日相伊藤博文等人会面图

HTML代码：

```
<table class="both-picture">
  <tbody>
    <tr>
      <td>
        <div class="duokan-image-single">
          <img alt="李鸿章1895年3月远赴东洋与日相伊藤博文等人会面图" src="../Images/lihongzhang1.jpg" zy-enlarge-src="self"/>
```

```
        </div>
      </td>
      <td>
        <div class="duokan-image-single">
          <img alt="李鸿章1895年3月远赴东洋与日相伊藤博文等人会面图" src="../Images/lihongzhang2.jpg" zy-enlarge-src="self"/>
        </div>
      </td>
    </tr>
    <tr>
      <td colspan="2"><p class="duokan-image-maintitle">李鸿章1895年3月远赴东洋与日相伊藤博文等人会面图</p></td>
    </tr>
  </tbody>
</table>
```

CSS代码：

```
table.both-picture {
    margin: .5em auto; /*并列图表格上下外边距为0.5em，水平方向设置居中*/
}
table.both-picture div.duokan-image-single {
    margin: 0; /*清除并列图中插图盒子的外边距*/
}
table.both-picture td {
    padding: 0; /*清除并列图中的单元格内边距*/
    border: 0 solid #333; /*清除单元格边框*/
}
table.both-picture td p.duokan-image-maintitle {
    margin: .25em; /*设置并列图中的图片说明外边距*/
    padding: 0; /*清除并列图中的图片说明内边距*/
    text-align: left; /*这个注释过长，居中显示不完整，所以设置对齐方式为居左*/
}
```

对于三张或以上尺寸一致的图片有另外一种排版——画廊模式。

10.4.4 画廊图

对于3张或以上尺寸一致的图片，推荐使用图片画廊模式。画廊模式下的图片，只会在阅读器中显示一张图片，其他图片隐藏。然后可以向左滑动切换下一张图片，如果有图片说明，图片说明也会跟着变动。

多看阅读——画廊模式

画廊模式可以支持在同一个位置显示多张大小一致的图像，用户可以通

过滑动等手势切换不同的图像内容。如下，即为一个拥有9帧画面的画廊（每一个duokan-image-gallery-cell声明一个分帧）：



狴犴

狴犴，其形似虎，有威力，故立于狱门上。



赑屃

貌似龟而好负重，有齿，力大可驮负三山五岳。



鸱吻

龙头鱼身，口润嗓粗而好吞。



蚣蝮

性喜水。



椒图

像螺蚌，性格好闭、好僻静。



蒲牢

受击就大声吼叫。



狻猊

形如狮，喜烟好坐。



饕餮

贪财为饕，贪食为餮。



睚眦

龙身豺首，性格刚烈，好勇擅斗。

HTML代码：

```
<div class="duokan-image-gallery">
  <div class="duokan-image-gallery-cell">
    <img alt="狴犴" src="../Images/longzi-bian.jpg"/>
    <p class="duokan-image-maintitle">狴犴</p>
    <p class="duokan-image-subtitle">龙生九子</p>
  </div>
  ...
  <div class="duokan-image-gallery-cell">
    <img alt="睚眦" src="../Images/longzi-yazi.jpg"/>
    <p class="duokan-image-maintitle">睚眦</p>
    <p class="duokan-image-subtitle">龙生九子</p>
  </div>
</div>
```

CSS代码：

```
div.duokan-image-gallery-cell img {
    margin: 0; /*清除画廊盒子中的图像外边距*/
    width: 100%; /*设置画廊盒子中的图像以预览窗口的100%宽度显示*/
}
p.duokan-image-subtitle {
    margin: 0 0 .5em; /*清除副标题的外边距*/
    font-family: "黑体"; /*定义副标题字体*/
    font-size: .8em; /*副标题字体大小*/
    text-indent: 0; /*清除缩进*/
}
p.duokan-image-maintitle {
    margin: 1em 0 0; /*图片说明的段间距*/
    font-family: "楷体"; /*图片说明使用的字体*/
    font-size: .9em; /*字体大小*/
    text-indent: 0; /*首行缩进为零，当你使用单标签p来指定首行缩进为2em时，记得在需要居中的文本中清除缩进，因为样式是叠加的*/
    text-align: center; /*图片说明水平居中*/
    color: #a52a2a; /*字体颜色*/
    line-height: 1.25em; /*行高，防止有很长的图片说明*/
}
div.duokan-image-gallery {
    margin: .5em auto; /*定义画廊图盒子的上下外边距*/
    width: 100%; /*定义画廊图盒子的宽度*/
    text-align: center; /*画廊图中的元素水平居中*/
}
```

掌阅iReader——轮播图

掌阅的画廊模式也称为轮播图，涉及到两个样式：zhangyue-c和zhangyue-img，分别定义div和img。CSS代码：

```
div.zhangyue-c {
    width: 100%; /*轮播图盒子宽度为预览窗口的100%*/
    margin: .5em auto; /*盒子上下外边距为0.5em，当图片比较小时，水平居中*/
    text-align: center; /*轮播图中的元素水平居中*/
}
img.zhangyue-img {
    width: 100%; /*设置轮播图盒子中的图像以预览窗口的100%宽度显示*/
}
```

HTML代码：

```
<div class="zhangyue-c" gallery="image">
  <div>
    <img class="zhangyue-img" zy-enlarge-src="none" src="../Images/longzi-bian.jpg" alt="狴犴"/>
```

```
    </div>
    <div>
        <img class="zhangyue-img" zy-enlarge-src="none" src="../Images/longzi-bixi.jpg" alt="赑屃"/>
    </div>
    <div>
        <img class="zhangyue-img" zy-enlarge-src="none" src="../Images/longzi-taotie.jpg" alt="饕餮"/>
    </div>
</div>
```

见10.4.2节添加「zhangyue-expansion.xml」文件。上面的代码在掌阅iReader阅读器中显示效果如下：

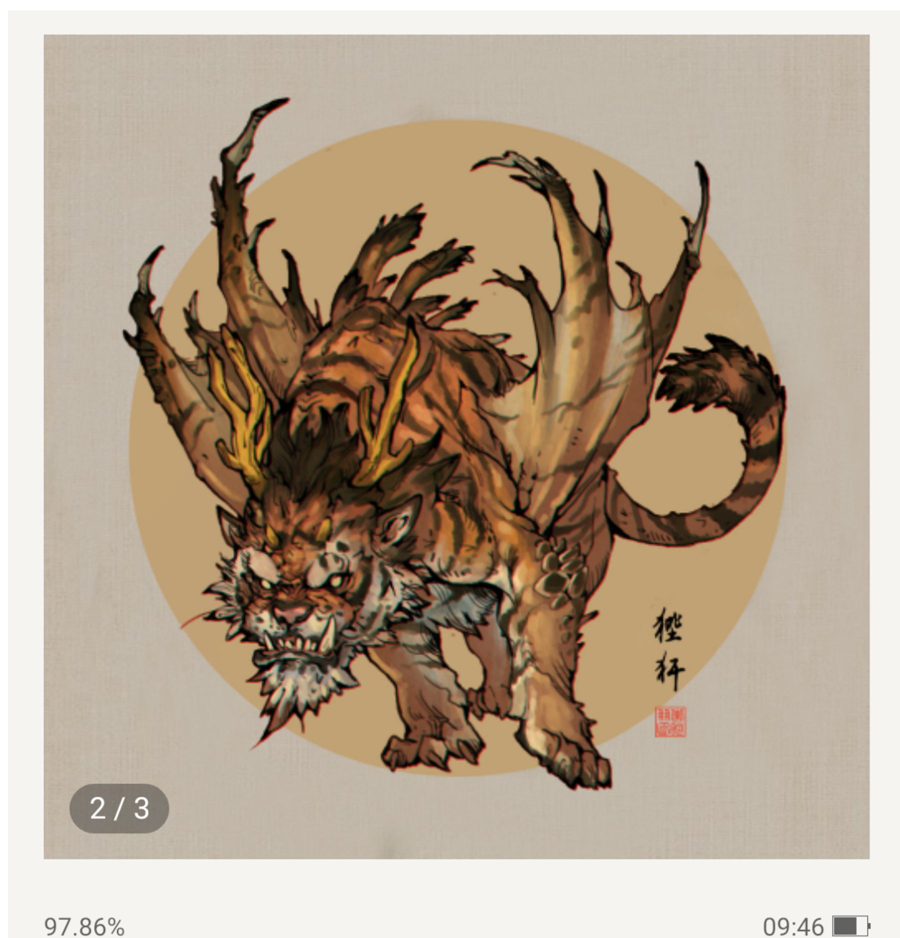


图10-12　掌阅的画廊模式

左右滑动即可切换图片。

10.4.5 行内图

在制作古籍的书中，有时会遇到一些生僻字，除了用内嵌字体的方法解决外，也可以使用图片代替。因为现在不少的阅读器都不支持MathML，所以数学公式或者一些化学公式也要用图片代替，插入到行中。

本书中的第5章，Sigil使用教程中也有不少图片是插入到行中的。

对于这种图片有的需要专门设置一定的宽度，使图片中的字体与正文字体接近；有的需要扣掉背景。总结之后如下：

- 设置宽度并去除背景：数学公式、化学公式；
- 去除背景：签名图、生僻字；

对图片的处理涉及到Photoshop的使用，具体细节可参考第6章。

生僻字

汉字都是方块形状的，宽高一致，所以CSS声明设置宽度或高度都是一样的。HTML代码：

```
<p class="bodycontent">生僻字<img class="rarefont" alt="rarefont-demo" src="../Images/rarefont-demo.png"/>是歛的类推简化字。</p>
<p class="bodycontent">生僻字<img class="rarefont" style="margin: 0 -0.25em .0625em -0.25em;" al
demo" src="../Images/rarefont-demo.png"/>是歛的类推简化字。</p>
```

CSS代码：

```
img.rarefont {
    margin: 0; /*清除外边距*/
    padding: 0; /*清除内边距*/
    height: 1em; /*高度为一个汉字的高度*/
    vertical-align: middle; /*垂直居中*/
}
```

多看阅读的行内垂直居中代码见各种样式类型内部的样式命令，显示效果：

生僻字是歛的类推简化字。

生僻字是歛的类推简化字。

在多看阅读、掌阅iReader客户端上，生僻字图片左右会有0.25em的空白，底部有0.0625em左右的下沉。所以如果是单独针对某个客户端做的

话，可以对生僻字样式增加外边距：

```
margin: 0 -0.25em .0625em -0.25em; /*清除多看、掌阅生僻字图片左右空白，并为图片添加底部外边距，提升字体*/
```

生僻字图片标准：

1. 尺寸：128px×128px
2. 格式：8bit的透明png
3. 反锯齿：无

数学公式

数学公式的CSS声明主要是设置宽度，而不是高度。同时在个别阅读器中注意设置为不交互效果。

HTML代码：

```
<p class="bodycontent">行内插入数学公式<img class="mathml-demo" alt="mathml-demo" src="../Images/mathml-demo.png"/>，以透明图片代替。</p>
```

CSS代码：

```
img.mathml-demo {
    margin: 0; /*清除外边距*/
    padding: 0; /*清除内边距*/
    width: 7em; /*为了显示不至于太大或太小*/
    vertical-align: middle;
}
```

显示效果：

行内插入数学公式 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ ，以透明图片代替。

公式图片大小设置时，可以在旁边打一个字母做参考，公式里的字符不能太大或太小。

签名图

签名图的CSS声明设置也是宽度，HTML：

```
<p class="inscribe">王菲<img class="autograph" alt="autograph" src="../Images/autograph.png"/></p>
```

CSS代码：

```
p.inscribe {
    margin: 1.5em 1em 0 0; /*上外边距和右外边距*/
    font-family: "楷体"; /*签名字体，可根据实际情况调整*/
    font-size: .9em; /*字体大小*/
    text-indent: 0; /*清除缩进*/
    text-align: right; /*水平方向右对齐*/
}
img.autograph {
    margin: 0; /*清除外边距*/
    padding: 0; /*清除内边距*/
    width: 4em; /*宽度*/
    vertical-align: middle; /*垂直对齐*/
}
```

显示效果：

王菲

以上就是3种行内图的排版案例。其中数学公式，有的比较长、有的比较短，单独写一个样式就有点多余了，可以考虑用内联式声明。同时在排版时注意公式图片大小及换行问题。

10.4.6 图片绕排

有时正文中的插图在竖直方向上较长，水平方向上较窄，如果使用居中显示的方式，这样会浪费显示空间也显得不美观。我们可以使用float属性，浮动插图单元到左侧或右侧。关于float属性，见3.4.11节。

一般左浮动的插图单元，只需要设置右外边距、下外边距；右浮动的插图单元设置左外边距、下外边距。

CSS代码如下：

```
/*左绕排*/
div.float-left {
    width: 50%;
    float: left;
    margin-right: .5em;
    margin-bottom: .5em;
}
/*右绕排*/
div.float-right {
    width: 50%;
```

```
    float: right;
    margin-left: .5em;
    margin-bottom: .5em;
}
div.float-right img {
    width: 100%;
}
```

下面演示一个图片的绕排：

就在今天，刚刚人民日报的编辑老师告诉我，我的句子入选了「一句话向祖国母亲表白」活动，将被人民日报印上北京地铁，带着「顾一宸」这个名字。

说实话，我很高兴，很兴奋！毕竟那可是人民日报，那可是北京地铁！



可更多的，我心里是笃定。因为，其他入选的作者是真的很有才华，而我不如他们有才，我唯有更加努力。

入选的其他作者可能都只花了几分钟想了一句或者两句话，而我当时足足花了一个小时，绞尽脑汁和乳汁，想了五句不同角度和语言风格的，都发了上去。

因为我是真的真的很想被人民日报印上北京地铁，但是那个微信群里都是上过人民日报公众号夜读栏目的作者，都是牛人，我不敢保证我的句子能够脱颖而出，所以，我就想着，你们就写一句，我特么写五句，付出五倍的努力，总有一句可以被选上吧。

果然，有志者，事竟成，苦心人，天不负，我入选了！

这正是我一直以来坚信并且坚持的一个道理，那就是：如果你真的

想要做成一件事情，很想很想，那么，就请你拿出120%的努力，尤其是一定要比别人更努力。

HTML代码：

```
...
<p class="bodycontent" style="margin-bottom: 1em;">说实话，我很高兴，很兴奋！毕竟那可是人民日报，那可是北京地铁！</p>
<div class="float-right">
  <img alt="three" src="../Images/three.jpg"/>
</div>
<p class="bodycontent">可更多的，我心里是笃定。因为，其他入选的作者是真的很有才华，而我不如他们有才，我唯有更加努力。</p>
...
```

在多看阅读器中浮动对象的前后，外边距叠加会消失。所以在浮动对象的上方元素中，添加内联样式下外边距，这样在显示上才能与全文中的段间距保持一致。

10.4.7　透明图昼夜自适应

昼夜自适应指的是，图片背景融于阅读界面背景。首先图片是透明png图片，在白天模式下，无论阅读界面背景是护眼模式、羊皮纸背景还是其他颜色的背景，插图都可以比较舒服的融于背景中，不会像右图，感觉插图很不舒服。

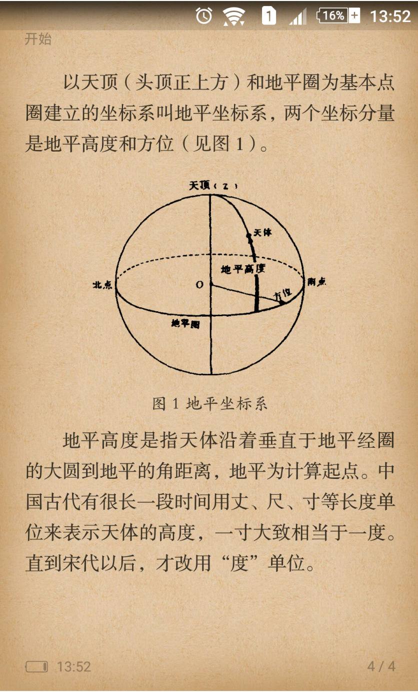


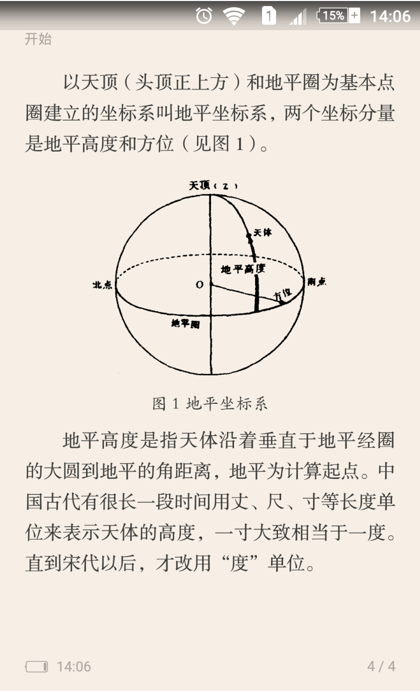


图10-13 插图背景融于阅读背景
图10-14 插图不融于阅读背景

不过插图背景如果是白色的，阅读界面背景也是白色，显示效果就不错，比较舒适。但是当切换到夜间模式下，黑白色的透明图显示效果就很差，基本上看不清插图了。所以我们需要一种模式，即可以在多种背景下融于阅读界面，又在夜间模式下可以清楚的显示黑白透明图片。这里就要用到多看的一个私有代码，代码如下：

```
<div class="duokan-image-static">
  <img alt="27" src="../Images/27.png"/>
  <p class="duokan-image-maintitle">图1 地平坐标系</p>
</div>
```

只要给图片容器引用duokan-image-static这个样式就可以实现上述功能。下面两张插图分别显示同一张插图在白天（左图）和夜间（右图）的显示效果：



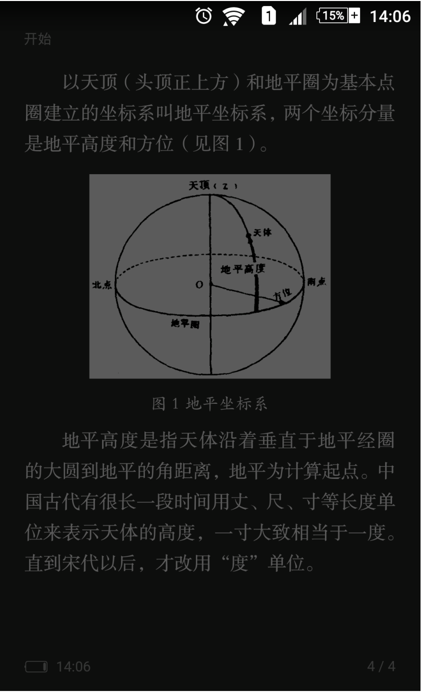
开始

以天顶（头顶正上方）和地平圈为基本点圈建立的坐标系叫地平坐标系，两个坐标分量是地平高度和方位（见图 1）。

图 1 地平坐标系

地平高度是指天体沿着垂直于地平经圈的大圆到地平的角距离，地平为计算起点。中国古代有很长一段时间用丈、尺、寸等长度单位来表示天体的高度，一寸大致相当于一度。直到宋代以后，才改用“度”单位。

14:06 4 / 4



开始

以天顶（头顶正上方）和地平圈为基本点圈建立的坐标系叫地平坐标系，两个坐标分量是地平高度和方位（见图 1）。

图 1 地平坐标系

地平高度是指天体沿着垂直于地平经圈的大圆到地平的角距离，地平为计算起点。中国古代有很长一段时间用丈、尺、寸等长度单位来表示天体的高度，一寸大致相当于一度。直到宋代以后，才改用“度”单位。

14:06 4 / 4

图10-16　夜间模式下的插图

> 阅读界面背景为护眼模式。本节原文参考了　@云轩阁主　的怎样制作昼夜自适应的插图电子书。

10.5　封面

封面全屏指的是在阅读器中一屏只显示封面图像，不显示其他内容。要做到这种效果，需要分别为不同的阅读器做适配。

10.5.1　多看阅读

在多看阅读器中实现全屏效果，需要从3方面设置。

图片：图片需要准备两张，一种16:9给手机，一张4:3给平板电脑。为手机准备的封面命名为*cover~slim.jpg*，为平板电脑准备的封面命名为*cover.jpg*。这样在不同的硬件中，阅读器会自动采用不同比例的封面。见6.3节。

HTML代码：

```
<body>
  <p><img alt="封面" src="../Images/cover.jpg"/></p>
</body>
```

也可以使用Sigil的「添加封面」功能添加默认的封面代码，效果和上面的代码一样，不影响全屏的显示。

OPF文件中添加一个扩展属性：

```
<spine toc="ncx">
  <itemref idref="Cover.xhtml" properties="duokan-page-fullscreen"/>
  ...
</spine>
```

id为「Cover.xhtml」的章会按全屏插图页逻辑处理添加这个扩展属性，在最后的验证阶段，会出现报错，可以不用管。

> HTML文件中应只含有一个img，不需要设置任何样式，程序会自动将图片撑满展示。

10.5.2 掌阅

掌阅的封面全屏与多看不同，它是需要添加一个xml文件才能生效。xml文件添加教程见10.3.2节中的掌阅部分。

图片要求：

1. 手机封面尺寸宽为1440px、高2560px，图片命名为*cover-phone.jpg*；
2. 平板封面尺寸宽为1536px、2048px，图片命名为*cover.jpg*。

掌阅对封面的全屏，不需要修改opf文件。而是为当前HTML添加图片背景，背景图片就是封面。对应的HTML代码：

```
<body style="background-image:url(../Images/cover.jpg);" class="background-img-center">
  <table style="width: 100%; height: 100%;">
    <tr style="height: 100%; text-align: right; vertical-align: bottom; zy-fontsize-adjust: fixed;">
      <td><img src="../Images/zy-cover-logo.png" style="width: 18%;" zy-enlarge-src="none" alt="" /></td>
    </tr>
  </table>
</body>
```

zy-cover-logo.png文件是掌阅的logo，可以不添加。样式background-img-center的代码：

```
body.background-img-center {
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center center;
    zy-fontsize-adjust: fixed;
}
```

掌阅全屏依赖*zhangyue-expansion.xml*文件，请务必添加！

10.5.3 Kindle

要求提供一个尺寸较大且分辨率较高的内容封面（意思是尽可能的大，但太大的图片也会占用Kindle宝贵的内容），因为如果内容封面太小，亚马逊将会拒绝您的发表申请（针对版权方而言）。

不要使用本节中所述方式以外的任何方式添加封面图片，否则可能会使封面在电子书中出现两次。

用下列方法之一在OPF文件定义内容封面：

方法1（首选）：

```
<manifest>
  ...
  <item id="cimage" media-type="image/jpeg" href="other_cover.jpg" properties="cover-image"/>
  ...
</manifest>
```

此语法是IDPF 3.0标准的一部分。

方法2：

```
<metadata>
  ...
  <meta name="cover" content="my-cover-image" />
  ...
</metadata>
...
<manifest>
  ...
  <item href="MyCoverImage.jpg" id="my-cover-image" media-type="image/jpeg" />
  ...
</manifest>
```

IDPF标准中并无这条句法规则。但是，该标准的制定却得益于IDPF的帮助，将通过IDPF验证程序进行验证。

亚马逊Kindle电子书发布指南：

```
英文版：https://kindlegen.s3.amazonaws.com/AmazonKindlePublishingGuidelines.pdf
中文版：https://kindlegen.s3.amazonaws.com/AmazonKindlePublishingGuidelines_CN.pdf
```

请复制链接在浏览器地址栏中粘贴，打开。

10.6 注释

注释是对图书内容或文字加以解释的总称。根据图书内容的性质，注释使用的符号和注释的位置有多种形式。如随文注、页下注、篇后注等。

注释的符号和格式

因为电子书中解决注释的最常见方案是弹注，即点击注释图标，在弹出框中显示注释内容。因此纸书中的注释符号在排电子书时可以不要。

但是有的图书内容不适合制作弹注，所以还是有必要了解常规注释的排版。

参考文献用「(1)、(1、3)」和「[1]、[1,5]」较多。注释符号用小五号排在左上角的位置。

用（注一）或［注一］做注码的，排在左下角的位置。

在竖排书籍中用中文扁数码，上下均加排括号，排在右上角。

注释的名称和位置

正文夹注。对正文中某一词语所作的诠释或对引文出处附带的说明，夹印于正文之内，称为正文夹注。其字体与正文相同，也可不同，常用圆括号断开。

随文注。注文排在需要注释词句的自然段下边。注释内容「齐右排」。一般距右版口两个字，句首破折号，回行齐肩。

脚注。亦称「边注」，「页末注」。脚注的内容除了与正文夹注相同外，还可对文中内容加以订正、补充或作阐发性的说明。

横排本的脚注排在本面版心下端，竖排本的脚注排在本面左侧，均用注线与正文隔开，注线与正文的距离不能大于「两行」。字号小于正文字号，行距小于正文行距。

被注文字按先后顺序编列注码在右上角，如①、②、③等，脚注序码必

须与其一致。脚注「注码序号」有两种形式，一种是每面另起序号；另一种是每节（章）另起序号，注意排版要求。

译校者所加的脚注，应在脚注文字结束句号后加一破折号，并分别写明「译者注」或「校者注」。

篇后注。文后注内容与脚注相同。集中编排在图书正文的某一部分后，可是篇后注、章后注、节后注，或是全部正文后的书后注。被注文字按先后顺序编码，文后注的序码必须与其一致。

呼应注。指某一词语在前文中已有注释，若再次出现，则以此注注明参见某页某注；或根据该词语在不同处用法的差异进行说明，各有侧重，互相呼应。

对正文表格中的某一项目需要特别说明时，也用呼应注，列在表的下方（或表内下方）。表中加注之处的右上角编码，如①、②等，呼应注的编码以本表为限。全表需要说明时，用「全表注」列于表的下方，前空两格写「注：」后接序号：1.、2.等（如只有一项全表注则可省略序号），再接写注义，转行齐肩。全表注编码以本表为限，若表中有全表注又有呼应注时，应先写全表注，后另起行写呼应注。

页下注注线的长度

页下注注线的长度是横版心宽的1/4或1/3长，可以使用百分比单位指定。

在设计注释时一般应掌握：

1. 要求版式美观。设计效果有利于读者查阅，注文与正文应有区别，又不妨碍正文的连贯性。
2. 用字号大小来区别正文与注文。一般正文夹注比正文字号小一号字，脚注比正文字号小两号字。
3. 用排版格式来区分正文与注文。注文行长比正文行长短，行空比正文行空窄；注文与正文之间用注线隔开；注释符号用角码、阴码、星花等标志，或用缩格、顶格、字空等形式区分。外文书中的注文有时排在切口空白处（边注）。
4. 用字体来区分正文与注文。例如正文用宋体，注文用仿宋体或楷体。外文书中正文用正体，注文用斜体。

电子书的注释解决方案

在实体书中，注释序号和注释内容是一起出现的，在一个页面中。如果电子书照搬其形式，不光是为排版带来问题，而且因为显示屏的尺寸限制，一页显示不了太多内容。现在普遍做法是为需要注释的字词或段落添加一个锚点，点击锚点弹出一个窗口，在窗口中显示注释的内容。这种方法称作弹出式注释，英文名Pop-up Footnotes。

现在各家阅读器对弹注的实现手段不尽相同，总的来说有4种方式：

1. 注释内容写在注释图片的alt属性中，有掌阅、当当、多看；
2. 使用传统的上下跳转的方式，在个别阅读器上实现弹注，电纸书Kindle；
3. 阅读器自身的扩展属性，多看阅读；
4. IDPF推荐的注释写法。

下面就以上4种方式详细说明注释的实现代码。

10.6.1　注释写在alt或扩展属性中

注释内容写在注释图片中的alt中或添加扩展属性存放注释内容，这种做法的缺点是注释内容只能是简单的文本内容，不可以有复杂的排版，比如插入图片、超链接、多行等。

掌阅

在制作掌阅版ePub电子书中，需要添加*zhangyue-expansion.xml*文件，这个文件的添加，可以在电子书已验证无问题后进行添加，添加方法见10.3.2节。

注释代码：

```
<img alt="" class="zhangyue-footnote" src="../Images/note-original.png" zy-footnote="注释内容" />
```

涉及的CSS代码：

```
img.zhangyue-footnote {
    width: .85em; /*注释图片宽度为正文字大小的0.8倍*/
}
```

排版样例：

9月5日是阳历年的第248天（闰年是249天），离一年的结束还有117天。

多看

多看阅读器也支持alt属性中的注释，实现代码：

```
<a class="duokan-footnote"><img alt="注释内容" src="../Images/note.png"/></a>
```

涉及CSS代码：

```
a.duokan-footnote img {
```

```
    width: .85em; /*<a>标签内的注释图片宽度为正文字大小的0.85倍*/
}
```

排版样例：

9月5日是阳历[注]年的第248天（闰年是249天），离一年的结束还有117天。

我们再看下掌阅的注释代码，综合多看，就可以做出一个兼容掌阅、多看的弹注。代码如下：

```
<a class="duokan-footnote"><img alt="注释内容" class="zhangyue-footnote" src="../Images/note-original.png" zy-footnote="注释内容"/></a>
```

给a元素的类可以不用再重复添加了！

10.6.2　超链接注释

在本书中的第5章中，讲过使用Sigil的插入ID和链接功能，制作注释。具体教程见5.4.10一节。

在电纸书Kindle设备中，直接采用上下跳转的方式即可以实现弹出式注释。

文中代码：

```
<a id="note1" href="#footnote1">[1]</a>
```

文下注释内容：

```
<p><a id="footnote1" href="#note1">[1]</a>正义或许会迟到，但永远不会缺席。</p>
```

如果你的注释内容不在同一HTML文件中，请给href属性添加具体的路径信息。

10.6.3　多看阅读弹注

多看阅读的弹出式注释除了上面的实现方式外，还有一种类似上下跳转的注释实现方式。这种注释方式可以实现比较复杂的排版。

在需要插入注释序号的文中插入如下代码：

```
<a class="duokan-footnote" href="#df-1"><img alt="" src="../Images/note.png"/></a>
<a class="duokan-footnote" href="#df-2"><img alt="" src="../Images/note.png"/></a>
<a class="duokan-footnote" href="#df-3"><img alt="" src="../Images/note.png"/></a>
```

在文章的末尾即</body>标签前插入注释内容代码：

```
<ol class="duokan-footnote-content">
  <li class="duokan-footnote-item" id="df-1"><p>这是一个注释文本1。</p></li>
  <li class="duokan-footnote-item" id="df-2"><p>这是一个注释文本2。</p></li>
  <li class="duokan-footnote-item" id="df-3"><p>这是一个注释文本3。</p></li>
</ol>
```

> duokan-footnote、duokan-footnote-content和duokan-footnote-item选择器不可以变更命名！
> 可以根据需要对duokan-footnote-content和duokan-footnote-item进行格式化，美化排版。

涉及的CSS代码：

```
a.duokan-footnote img {
    width: .85em; /*<a>标签内的注释图片宽度为正文字大小的0.85倍*/
}
```

排版样例：

9月5日是阳历年的第248天（闰年是249天），离一年的结束还有117天。

再给注释内容格式化一下：

```
/*多看注释内容*/
li.duokan-footnote-item p {
    text-indent: 0; /*如果你是使用单标签p来设置正文格式，需要添加这个声明去除首行缩进*/
    font-family: "仿宋"; /*注释正文字体*/
    font-size: .85em; /*注释正文字号比正文字号要小一些*/
    line-height: 1.35em; /*行高*/
    text-align: justify; /*两端对齐*/
}
/*多看注释盒子*/
ol.duokan-footnote-content {
```

```
    border-top: 1px #000 solid; /*模拟纸书，在第1个注释上面添加一条横线，以与正文隔开*/
    margin-top: 1.5em; /*注释内容与正文的间距*/
}
```

上面的效果可以在Sigil的编辑窗口种可以看到，在阅读系统种就是上面的显示效果。

> `<a>`标签的内容必须为图片，文字无效。
> 标注列表必须为`<ol>`标签列表，`<ul>`标签无效。如果你不希望注释内容出现编号，可以使用`list-style-type`属性更改它。

Kindle和多看阅读兼容注释

这个方案是 @林可可 总结的代码，已取得转载许可。代码如下：

注释上标

```
<sup><a class="duokan-footnote" style="text-decoration: none!important" id="note-1" href="#footnote-1"><img alt="注释iccon" src="../Images/note.png"/></a></sup>
```

注释内容

```
<ol class="duokan-footnote-content">
  <li class="duokan-footnote-item" id="footnote-1"><p class="footnote"><a style="text-decoration: none!important" href="#note-1">◎</a>参见作者2004年出版的《球状闪电》。</p></li>
</ol>
```

加`text-decoration`属性是为了清除在Kindle中的超链接下划线。

据 @PartyPat 读者意见修改如下：

在`<sup>`标签默认样式`font-size: smaller;`下，注释图标在iBooks macOS中显示过小。建议：

1. 在CSS中重新对`font-size`属性定义，比如`font-size: 0.55em;`；
2. 注释标号处不使用`<sup>`标签，给`<a>`标签设置`vertical-align: super;`样式，并定义注释图标大小，比如`width: 0.85em;`。

另外，关于兼容注释，再次重申一下，注释只适配一个阅读器，换一个阅读器，就改下代码。不然，注释代码变得臃肿难读，且会影响弹出注释的展示。

10.6.4 ePub3标准注释

在传统纸质出版物中，footnote是指存在与本页面正文下方的注释，我们阅读纸质书时一般见到的都是这种注释：在正文中标记注释符号，在正文所在页下方放置注释内容。ePub 3标准的footnote写法如下：

```
<a epub:type="noteref" href="#footnote_1">注1</a>
<a epub:type="noteref" href="#footnote_2">注2</a>
<a epub:type="noteref" href="#footnote_3">注3</a>
```

上面是插入到文中的注释序号，`epub:type="noteref"`来触发弹注。

```
<section epub:type="footnotes">
  <aside epub:type="footnote" id="footnote_1"><p>这是注释内容1。</p></aside>
  <aside epub:type="footnote" id="footnote_2"><p>这是注释内容2。</p></aside>
  <aside epub:type="footnote" id="footnote_3"><p>这是注释内容3。</p></aside>
</section>
```

这是插入到`</body>`标签前的注释内容，`epub:type="footnote"`表示脚注的文字。文件头更改：

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops">
```

> 只有ePub 3标准才可以使用`epub:type`规则，ePub 2的电子书需要在`<html>`标签的命名空间添加`xmlns:epub="http://www.idpf.org/2007/ops"`，才可以让注释弹注。

排版样例，请使用Lithium或Gitden阅读器打开查看：

太平广记

小说总集。北宋李昉等编辑。因书成于宋太宗太平兴国年间，故名。五百卷，另目录十卷。按性质分九十二大类。采录自汉至宋初的小说、笔记、稗史等四百多种，保存了大量的古小说资料。其中引用的书，有很多已经散佚、残缺或被后人窜改，赖此书得以考见。但所采各书文字，亦多有所改动。有明嘉靖间谈恺刻本和今人校正排印本。

朝代名。宋太祖赵匡胤于公元960年建宋王朝，到钦宗靖康二年（公元1127年）止，皆建都汴京（今河南开封市），共历九帝，统治一百六十七年，史称北宋。

人名。（公元925年～996年）字明远，宋深州饶阳人。太宗时，任平章事。端拱年间，边境告急，李昉引汉唐故事，以屈己、修好、弭兵、息民为言，受时人称道。其性温厚，好接宾客。为文慕白居易，浅近易懂。卒谥文正。曾奉敕编纂《太平御览》、《太平广记》、《文苑英

华》等。

宋太宗（公元939年 ~ 997年）：即「赵炅」。北宋皇帝。原名匡义，后改光义，即位后改炅。太祖弟。继位后兼并吴越，灭北汉，结束了五代十国分裂割据的局面。在对辽的两次战争中遭大败。政治上继续加强专制主义中央集权，禁止节度使补亲吏为镇将，收节度使所领支郡由朝廷直辖。设考课院、审官院，加强对官吏考查与选拔，又大量增加进士科名额，编修《太平御览》等书，加强「重文」风气。

太平兴国（公元976年 ~ 984年）是北宋君主宋太宗赵匡义的一个年号，共计近8年。吴越忠懿王钱俶亦用该年号纪年（976年十二月 - 978年五月）。

嘉靖（公元1522年 ~ 1566年）是明世宗的年号，明朝使用嘉靖这个年号一共45年，是明朝使用第二长的年号。嘉靖，出自《尚书·无逸》中「嘉靖殷邦」之语。明世宗即位时内阁备选年号第一为「明良」、其次为「绍治」、第三为「嘉靖」。

谈恺（公元1503年 ~ 1569年），字守教，号十山。无锡（今江苏省无锡市）人。谈恺为谈信十世孙。嘉靖四年（1525年）中式乙酉科举人，嘉靖五年（1526年）联捷丙戌科进士，授官户部员外郎，「五迁而后为四川副使」。晓军事，曾平定江西土匪李文彪和海寇徐壁溪，因平乱有功，拜兵部右侍郎，官至都察院都御史。嘉靖四十五年（1566年），谈恺据《太平广记》传钞本加以校补重印，称《谈刻本》，是今存最早的版本。著有《前后平粤录》四卷、《虔台续志》五卷等书。有子谈志伊。

10.6.5 汉字注音

汉字注音在2.2.11节描述的比较清楚，可参考该节处理中文的汉字注音。因第2章标题名不突出，故重列一节以兹说明。

床（chuáng）前（qián）明（míng）月（yuè）光（guāng），疑（yí）是（shì）地（dì）上（shàng）霜（shuāng）。

举（jǔ）头（tóu）望（wàng）明（míng）月（yuè），低（dī）头（tóu）思（sī）故（gù）乡（xiāng）。

关于拼音字体这两天（2018年5月2日左右），又知道了两款拼音字体，一个是GB Pinyinok，另一个是Chinese Pinyin。GB Pinyinok是在《排版基础知识》中，在第二节提到《新华字典》的拼音就是用了这款字体。Chinese　Pinyin这款字体是在Ps中搜索查到的，应该是系统默认安装的。

GB Pinyinok效果：

床（chuáng）前（qián）明（míng）月（yuè）光（guāng）

Chinese Pinyin的效果不理想，就不提供示例了。

---

1. 这个内容，在Sigil中是可以被编辑的，而在多看阅读系统中默认隐藏，只有被激活后才会显示。

10.7　多媒体

在2.1.11和2.1.12二节已经讲过标准多媒体代码的写法，虽然在Gitden和Lithium阅读器中可以正常播放，不过在多看阅读和掌阅中就不可以了。这种情况一般是对ePub规范支持不充分引起了，但是有的电子书又有这种插入音频和视频的需求。所以针对多看阅读和掌阅，它们都有基于自身阅读器的扩展属性。

10.7.1　音频

建议MP3中采用立体声声道，必要时尽可能调高比特率，以听清楚音频内容。要有良好的视听效果，请考虑将比特率设置在128至320kbps（千比特每秒）之间。

多看阅读

在HTML 5规范中，音频采用标准的audio标签，但需要扩展说明其占位图像，示例如下：

```
<audio controls="controls" class="duokan-audio content-speaker" placeholder="../Images/play-before.png" activestate="../Images/play-after.png" title="虎吼">
  <source src="../Audio/Tiger.mp3" type="audio/mpeg"/>您的浏览器不支持audio标签。
</audio>
```

duokan-audio为扩展标签，表明了该audio标签为多看扩展类型，`placeholder`用于表示占位图，未播放时，`activestate`表示活动状态下的占位图，正在播放时，`title`表示标题名。

audio的`controls`属性出现时，表明应用层需要显示控制栏，如果不出现，则无需显示控制栏。

一般情况下可以指定audio采用的CSS样式，在绘制占位图时需要遵循该

样式，示例代码如下：

```
audio.content-speaker {
    width: 4em;
}
```

排版样例：

您的浏览器不支持audio标签。

掌阅

掌阅的音频代码与多看不同，只需要给音频添加一个封面。照例，仍需要添加*zhangyue-expansion.xml*文件。

```
<audio class="zhangyue-audio" controls="controls" src="../Audio/Tiger.mp3" zy-cover-pic="../Images/Tiger.png">您的浏览器不支持audio标签。</audio>
```

涉及CSS代码：

```
audio.zhangyue-audio {
    width: 3.5em; /*定义占位图的宽度*/
}
```

10.7.2　视频

由于视频内容中可能包含音频内容，建议尽可能在音频源中使用立体声通道。

多看

在HTML 5规范中，视频采用标准的video标签，示例如下：

```
<video class="duokan-video content-matrix" poster="../Images/sunup.jpg">
  <source src="../Video/sunup.mp4" type="video/mp4"/>您的浏览器不支持video标签。
</video>
```

duokan-video为扩展标签，表明了该video标签为多看扩展类型。

一般情况下可以指定video采用的CSS样式，在绘制`poster`时需要遵循该样式，示例代码如下：

```
video.content-matrix {
```

```
    width: 540px;
    height: 304px;
}
```

video的`controls`属性禁止出现。

排版样例：

您的浏览器不支持video标签。

掌阅

掌阅的视频代码：

```
<video class="zhangyue-video" src="../Video/sunup.mp4" zy-cover-pic="../Images/sunup.jpg">您的浏览器不支持video标签。</video>
```

涉及的CSS代码：

```
video.zhangyue-video {
    width: 100%;
}
```

10.8 表格

表格与插图一样是图书内容另一种表达形式。表格具有条理清楚、对比性强等优点。表格版式分为两大类，一类是零件表格（属于填写性质的表格），另一类是书版表格（属于表达式或者说是表示性的表格）。表示性表格要求格式符合规范，数字表格、文字表格表达清楚，栏宽疏密适中、合理。

表格的内容和工艺的分类。

a. 以表格内容分。数据表、归纳表、系统表、会计表、统计表等；
b. 以表格形式分。全线式、省线式、联线式（流程表）、方框式（系统表）；
c. 以表格用字分。文字表、数字表、公式表等。

表格的排版原则。

a. 表格位置。表格位置和插图相同，先见文后见表，表随文走的原则。
b. 表格宽度。宽度有两种形式，如果全书表格的内容较多，表格宽度采取屏幕全宽度的版式；如果全书表格的内容较少，表格宽度可采取全书统一左右缩格的版式。一般在平板电脑上采用浮动，手机都建议用全宽度的办法处理。
c. 表格用线。表格的心线用全线或者是用省线以及表格四周的框线形式，全书应该一致。

区分表格项和内容。绝大多数表格项位于表格的上方和左边，易读的表格能使读者一目了然地分清表格项和内容。我们可以通过加粗划分单元的网格线或为标题加上底色来达到这一目的。

使内容便于理解。用表格来呈现数值能方便读者理解和阅读，因此表格最重要的一点就是要简单易读。为此我们要下点功夫，比如为表格的行或列加上颜色，使读者在比较数值时不至于看向别的单元。

对齐位数。表格里的数值都是具有意义的，必须要使读者能清晰地辨认出它。因此在制作时要保证个单元中小数点的位置和数字的位数是统一对齐的。

明确区分表格项和内容，用交替的底色来归类各列、各行，使表格更易于理解。

表格的结构。

由表题、表头、表身和表尾（表注）组成。表格的结构是指表格各部分的排版」术语」（见下图）。

表格框线的版式。

框线用「反线」。框线有三种版式：四框排线；上下排线，左右不排线；上排线，左右下不排线（俗称三开口）。

表格心线的版式。

心线用「正线」。分为栏线、行线全排（全线表）；只排栏线不排行线（俗称省线表）；只排顶线、表头下线、底线（俗称三线表、三线表不排左右墙线）三种版式（见下图）。

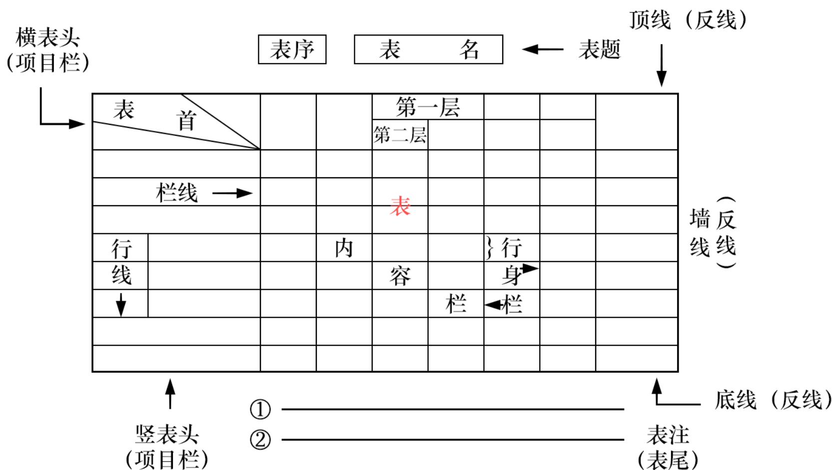


图10-17　表格的各部位名称

全线表。行线、栏线全排（三种框线任选其一）。

项目	甲	乙	丙	
丁	戊			
A	1	4	7	10
B	2	5	8	11
C	3	6	9	12

省线表。只排栏线不排行线（三种框线任选其一）。

项目	甲	乙	丙	
丁	戊			
A	1	4	7	10
B	2	5	8	11
C	3	6	9	12

三线表。只排表头下线（框线可选三开口或上下线）。

项目	甲	乙	丙	
丁	戊			
A	1	4	7	10
B	2	5	8	11
C	3	6	9	12

原稿表格的框线和心线由于编辑的原因，框线和心线可能是一个表格一种形式，这样影响全书版式的美观、规范，所以必须对全书表格线型的要求作出说明。

表格的表头高度、横格高度和无行线表格行距的版式。

①表格的表头高度大于表格横格的高度（以区别于表格内容栏）。

②两层表头每层两个字（表头内容较多可适当加高）。

③表身横格为两个字高。

④表身的行距（省线表）：表格行距为表格字号的1/2。

A.5行以下行距1个字，表格内容上下距线一个字。

B.6行以上行距1/2（表格字号），表格内容上下距线一个字。

⑤为了调整版面，横格高可以适当加高（3-5个字）。

⑥为了调整版面，行距可以适当加宽（2-3个字）。

表格横表头的格式。

表格横表头和竖表头同是表格的「项目栏」，等同于正文的「小标题」，格式必须规范才能清楚、美观。

横表头版式是以项目格的宽度为准，项目内容第一个字至最后一个字中间的宽度是项目内容距左右栏线距离的「和」，如表格格宽八个字，项目是两个字，两个字中间字距三个字，左右距栏线一个半字；项目是三个字，三个字中间字距一个字，三个字左右距栏线一个半字。如果表格横表头项目字数较多，采用文字叠排左右撑齐的格式。

表格竖表头的格式。

a. 表头内容的字数有规律。内容左右撑齐（如：电视机、洗衣机、冰箱）。
b. 表头内容的字数长短不等。不足四个字，加开至四个字；超过四个字不加开；表头的内容需要回行，回行低一格。
c. 表头内容属于子、目式。用条款式（阶梯式）格式。

数字表内容栏的格式。

①数位内容差别不大的数字要对位排；数位内容差别较大的（如1,500，0.75）数字后齐排。

②数字在栏中的位置：一种是居中偏后排；一种是距右栏线固定字数排（如距右栏线对开，一字）。

文字表的格式。

文字表格有三种格式。由于表格内基本都是文字，根据表格的项目和内容的多少不同，各栏应采取不同的格式。

起行顶格回行低格：用于竖表头和单行文字较多的栏。

起行低格回行顶格：用于表格的表身（内容栏）。

起行顶格回行顶格：用于五倍以下的窄栏。

表格的几种改排方法。

原稿表格尺寸的大小是根据其内容而定的，形式和大小如果在设定的版心上排不下，则需要改排处理。需要处理的表格必须严格按照规范进行，否则将给读者造成阅读不便，甚至内容表达有误。改排有竖折栏、横折栏等。

竖折栏（折栏）。当表格横向的内容少，竖向的内容长或受版心版式的限制，可改为「折栏排」；折栏的部分加排横表头，栏与栏之间加排双线。

折栏表格可以折两栏以上的多栏表格。如果原稿是多折栏的表格，由于版心的限制可改为「通栏表格」。

横折栏（叠排）。表格横向的内容长超出横版心，也超出竖版心；竖向的内容不超出竖版心的1/2，可改为「叠排」；

叠排部分加排竖表头，上下层之间加排双线。

叠排表格也可以叠多层；为适应调整版心的需要，也可以在叠排表格的表头上加排「续表」两字。

如果多看阅读器支持的选择器更丰富一些，排版表格是比较简单的事，但是现在只能使用class的方式选择要排版的对象。所以，表格的排版尽量要简单一些。

10.9 竖排

中华人民共和国成立后，于实施汉字简化政策之前，已逐渐推广印刷品文字横排。1955年1月1日起，《光明日报》等率先改为横排，同年12月30日文化部发布《关于汉文书籍、杂志横排的原则规定》；1956年1月1日起，《人民日报》等中央、地方报一律改为横排。竖排文字一直到1980年代早期还在中国少数出版物（如某些连环画）中使用。目前基本只有新印古籍、学术著作和现当代人的古体诗文集还使用竖排，而引进的日本漫画至今还使用竖排。

中国大陆和新加坡推行简体中文之后，竖排文字已经不多见于出版物，但在排版空间受限的场合会使用，如书脊、标题等。早期的简体中文出版物有时会采用竖排，但现今则几乎都是横排的。有些简体字书报以横排为主，但部分版面会使用竖排。传统文化内容，如大部分书法作品（硬笔书法除外）、请帖和全部对联即使混用繁、简体字时仍然直写，排印出版的古籍大多数也仍用繁体字竖排（如果采用简化字，则极少竖排）。

表26　竖排与横排之比较

排版	书写方向	翻页方向	代表
竖排	自上而下↓	自左而右→	英语、简体中文……
横排	自左而右→	自右而左←	繁体中文、日文……

当你确定要用竖排时，需要给文中的标点符号替换为繁体的标点符号。

表27　竖排与横排标点的使用

	台湾	中国大陆	香港 澳门	日本
逗号 句号	，。 置于中间	，。 横排时置于左下 竖排时置于右上	，。 （无严格规范）	横：、。／，。 置于左下 纵：、。 置于右上
叹号	！ 置于中间	！ 横排时置于左边 竖排时置于右边	！ （无严格规范）	！ 置于中间
顿号	、 置于中间	、 横排时置于左下 竖排时置于右上	、 （无严格规范）	·（中黑） 置于中间
冒号	：	：	：	：

	置于中间	横排时置于左边 竖排时置于右边	（无严格规范）	横排时置于中间 竖排时旋转90°
省略号	…… 置于行中 竖排时旋转90°	同左 （无严格规范）	…或…… 置于行中 竖排时旋转90° （无严格规范）	
引号	横排：「『』」 竖排： □ □ □ □	横排：「『』」 竖排： □ □ □ □		横排：「『』」 竖排： □ □ □ □ （无严格规范）
着重号	. 横排时置于字下， 竖排时置于字右	. （无严格规范）		
专名号 下划线	＿ 横排时置于字下， 竖排时置于字左	＿ （无严格规范）		

竖排书的另行（段）

一般有三种方式。接近现代的是第一种版式；仿古的是第二种和第三种版式。

①起行低两格，回行顶格。

②起行顶格，回行低格。

③起行顶格，回行顶格。

竖排书的夹注

古籍书注释除去「篇后注」和「页末注」以外，还有夹注的注释版式。

注释内容排在正文中需要解释的字和词下「双行排」。注释的字号小于正文字号，注释的双行大于正文字号。双行夹注在ePub中还实现不了。

竖排的实现需要阅读器支持，下面介绍3种方法分别针对不同的情况。

10.9.1　支持ePub3的阅读器

属性`writing-mode`设置文字的书写方向，最新的规范中缩减为三个值：

1. horizontal-tb（默认值）：自上而下，从左到右的横排书写方式。
2. vertical-rl：从右到左，自上而下的竖排书写方式（典型的汉字竖排排版方式）。
3. vertical-lr：从左到右，自上而下的竖排书写方式（主要用于内蒙古的蒙古语和满语）。

HTML代码：

```
<body class="vertical-mode">
    <p>中国古代文献多是竖排，主要是因为上古书写材料多为竹片木条，文字竖直
```

```
写上去，然后一条条横向平行排列串成长卷，这样有利用使用左右手展开阅读。直到后来出现纸质书，这一排版习惯一直传承下来。五四期间，打倒一切的先行者们，事事要向先进的西方看齐，自然也要拿竖排书来作文章。</p>
</body>
```

CSS代码：

```
body.vertical-mode {
    -webkit-writing-mode: vertical-rl; /*兼容Chrome浏览器*/
    -epub-writing-mode: vertical-rl; /*epub扩展*/
    writing-mode: vertical-rl;
}
```

这里需要注意的是，竖排样式一定是给body元素的。

然后在OPF文件中设置翻页顺序：

```
<spine toc="ncx" page-progression-direction="rtl">
  ...
</spine>
```

10.9.2　多看阅读

在多看阅读中，给spine元素设置翻页方向为从右到左，即可以实现竖排效果，同时简体中文的标点符号也会自动更改为繁体中文形式。至于CSS则不需要进行任何变动，在Sigil的编辑界面显示的仍然是横排效果。

打开OPF文件，添加下面的代码：

```
<spine toc="ncx" page-progression-direction="rtl">
  ...
</spine>
```

`page-progression-direction`设置出版物内容流动的全局方向。可设置的值有ltr（从左到右）、rtl（从右到左）和default。

10.9.3　浮动方式实现竖排

使用浮动的方式实现竖排，要点是给浮动元素设置一定的宽度。

春
江

花月夜

春江潮水连海平
海上明月共潮生
滟滟随波千万里
何处春江无月明
江流宛转绕芳甸
月照花林皆似霰
空里流霜不觉

飞

汀上白沙看不见

竖排时，应避免段落缩进，不然会出现每行均换页的蛋疼效果，一般可以用`text-indent: 0em;`来避免；但若是opf中增加了duokan-book-id，则会具有默认段落自动缩进效果，这时需要额外增加多看扩展声明`duokan-text-indent: 0em;`来清除duokan-book-id带来的自动缩进。感谢书友@lanjing512 指正！

自己制作的电子书，在多看阅读中阅读的话，会进行一些设置，方法见自制书阅读设置。

浮动之后需要在浮动元素后面添加清除浮动样式，避免影响下文。

HTML代码：

```
<p class="poetry_center-title">春江花月夜</p>
  <div class="poetry_center-text">
    <p>春江潮水连海平</p>
    <p>海上明月共潮生</p>
    <p>滟滟随波千万里</p>
    <p>何处春江无月明</p>
    <p>江流宛转绕芳甸</p>
    <p>月照花林皆似霰</p>
    <p>空里流霜不觉飞</p>
    <p>汀上白沙看不见</p>
  </div>
 <div style="clear: both;"></div>
</div>
```

CSS代码：

```
p.poetry_center-title {
    margin: .5em .875em .25em .5em;
    font-family: "小标宋";
    font-size: 1em;
    width: 1em;
    float: right;
    color: #da251d;
}
```

```
div.poetry_center-text p {
    margin: .5em .25em 0 0;
    line-height: 1.3em;
    font-family: "仿宋";
    font-size: .875em;
    width: 1em;
    float: right;
}
```

这种竖排的缺点是要求有一个合适的显示区域，不然竖排的分成两屏显示；还一个缺点是标点的问题，所以只适合对标题的排版。

10.9.4　Kindle的竖排问题

Kindle的内置繁体字体有两种：宋体（Song　T）、黑体（MYing　Hei　T）。比如正文是繁体，给正文设置如下样式：

```
p {
    margin: .5em 0;
    line-height: 1.5em;
    text-indent: 2em;
    text-align: justify;
    font-family: "Song T"; /*指定使用宋體（繁體）*/
}
```

如果要用黑体（繁体），直接更改「Song　T」为「MYing　Hei　T」即可。下面的排版案例我采用了一个竖排的小技巧。HTML代码如下：

```
<p class="title-text" lang="zh-hant">我很好啊
<span class="punctuation">，</span>妈</p>
```

涉及到CSS样式：

```
p.title-text {
    margin-right: 10%;
    float: right;
    font-size: 1.25em;
    text-indent: 0;
    -webkit-writing-mode: vertical-rl; /*兼容Chrome浏览器*/
    -epub-writing-mode: vertical-rl; /*epub扩展*/
    writing-mode: vertical-rl;
}
/*竖排标点*/
span.punctuation {
    font-family: "Song T";
```

```
}
```

在Kindle电子书中显示如下：

Chapter 1

我很好啊，妈

全书还剩：5时 8分 7%

图10-18　Kindle的竖排篇章页

可以看到标点是显示在中间的，而不是横排时偏左。

以上是只用Kindle提供的宋体来排版，但是有时我们也会用来别的字体来排某些特殊段落，比如诗词、引文等，竖排排版效果如下：

小兒養得嬌，自來生得腼腆，不肯對人胡亂便唱。」于是西門慶便叫玳安書袋內取出五兩一錠銀子來，放在桌上，說道：「這些不當甚麼，權與桂姐為脂粉之需，改日另送幾套織金衣服。」

桂姐連忙起身謝了。先令丫鬟收去，方纔下席來唱。這桂姐雖年紀不多，却色藝過人，當下不慌不忙，輕扶羅袖，擺動湘裙，袖口邊搭剌着一方銀紅撮穗的落花流水汗巾兒，歌唱道：

駐雲飛 舉止從容，壓盡拘攔占上風。行動香風送，頻使人欽重。

嗏！玉杵污泥中，豈凡庸？一曲宮商，滿座皆驚動。勝似襄王一夢中，

勝似襄王一夢中。

唱畢，把個西門慶歡喜的沒入脚處。分付玳安回馬家去，晚夕就在李桂卿房裡歇了一宿。緊着西門慶要梳籠這女子，又被應伯爵、謝希大兩個一力攛掇，就上了道兒。次日，使小厮往家去拏五十兩銀子，段舖內討四件衣裳，要梳籠桂姐。那李嬌兒聽見要梳籠他的姪女兒，如何不喜？連忙拏了

本章还剩：1分　　21%

图10-19　仿宋字体的诗词排版

仔细观察上面的排版，可以看到在仿宋段落中，标点处于左下角位置，而不是和正文宋体那样，在中间的位置。那么我们现在要解决的问题就是，让文字显示为仿宋，标点符号显示为竖排繁体标点。

这里需要用到一个CSS属性`unicode-range`，来实现字体的混搭。这个方法是利用了font-family属性的备选方案原理，示意如下：

```
/*方正屏显书宋_GBK 版本1.00*/
@font-face {
    font-family: "宋体";
```

```
    src: local("st"),local("SongTi"),local("Songti"),
         local("宋体"),local("明体"),local("明朝"),
         local("HYMyeongJo"),local("DK-SONGTI"),
    url("../Fonts/YaSong.ttf");
}
```

它会先查找名称为「st」的字体，如果该字体不存在，则查找「SongTi」，以此类推，直到找到可用字体。

利用此原理，我们可以用unicode-range属性指定一个字体中的一部分字符集，然后再定义一个标点符号的字体，来实现我们需要的效果。

HTML代码如下（如竖排、其他样式略）：

```
<p>【駐雲飛】舉止從容，壓盡勾欄占上風。行動香風送，頻使人欽重。嗏！玉杵污泥中，豈凡庸？一曲宮商，滿座皆驚動。勝似襄王一夢中，勝似襄王一夢中。</p>
```

CSS样式：

```
@font-face {
    font-family: "仿宋";
    src: local("仿宋"),local("DK-FANGSONG"),url("../Fonts/FangSong.ttf");
    unicode-range: U+4E00-9FA5,U+3400-4DB5;
}
p {
    font-family: "仿宋","Song T";
}
```

其中`U+4E00-9FA5`指的是中日韩统一表意文字；`U+3400-4DB5`指的是中日韩统一表意文字扩展A。实际显示的效果：

桂姐，因此同西門爹送回來。快看酒來，俺們樂飲三盃。」虔婆讓三位上首坐了。一面點茶，一面打抹春擡，收拾酒菜。少頃，掌上燈燭，酒餚羅列。桂姐從新房中打扮出來，旁邊陪坐，免不得姐妹兩個金樽滿泛，玉阮同調，歌唱遞酒。正是：

琉璃鍾，琥珀濃，小槽酒滴珍珠紅。烹龍炮鳳玉脂泣，羅幃繡幙圍香風。吹龍笛，擊鼉鼓。皓齒歌，細腰舞。況是青春莫虛度，銀缸掩映嬌娥語，不到劉伶墳上去。

當下姊妹兩個唱了一套，席上觥籌交錯飲酒。西門慶向桂卿道：「今日二位在此，久聞桂姐善能歌唱南曲，何不請歌一詞，奉勸二位一盃兒酒！」應伯爵道：「我又不當起動，借大官人餘光，洗耳願聽佳音。」那桂姐坐着只是笑，半晌不動身。原來西門慶有心要梳籠桂姐，故先索落他唱。那院中婆娘見識精明，早已看破了八九分。桂卿在旁，就先開口說道：「我家桂姐從小兒養得嬌，自來生得靦腆，不肯對人胡亂便唱。」于是西門慶便叫玳安書袋內取出五兩一錠銀子來，放在桌上，說道：「這些不當甚麼，權與桂姐為

本章还剩：1分　　21%

图10-20　楷体竖排

在上面的排版中，仿宋段落表现正常；楷体段落却有点偏左，但比之前的已经好很多了。

感谢 @王萌萌 提供10-19、10-20、10-21三张图片及思路，本节也参考了 @张鑫旭 大神的教程——CSS unicode-range特定字符使用font-face自定义字体。

10.10 生僻字

阁主有篇文章介绍了ePub中生僻字的处理，对常见的几种方法都做了描述。但无具体的操作步骤，本节据此补充了详细的操作细节。

10.10.1 替换

查找一个字异体字的方式除了使用字典外，还可以使用一些文字工具网站来查找。这比翻字典快多了。我经常使用的网站是国学大师网，查找一个字的方式除了直接输入该字以外，还可以使用部件查字的方式查找。

比如我要查找𬸦这个字的，这个字所在平面为E区，普通输入法无法打出这个字，使用输入法的u模式也不行。查《通用规范汉字字典》，是个三级字，用于人名，对应的繁体字为鷟。这个字是在基本区，一般字体都能显示。知道了这个信息之后，就方便多了。可以直接输入鷟字，网站会提供该字的异体字。像这种，繁体在基本区，简化字在扩展区的有不少。回到部件查字中，我们就可以把这个字拆分，直接输入「族鸟」，然后搜索。在查询结果列表中，找到我们要的这个字，在详情页中找到它的异体字。然后复制，粘贴到要用的文字处。

除了在国学大师网可以查字外，还可以在汉典、古今文字集成等网站查字。如何使用部件查字，在网站中都有教程。

10.10.2 偏旁组合

汉字种类多，字的笔画有多有少，最少的仅有一笔，最多可以有三十多笔。笔画之间存在大小、高低、长短、疏密的关系，而按照空间结构主要有上中下、左中右之分。汉字字形可分独体字、合体字两大类，偏旁是合体字的结构单位。

Unicode中提供了12种文字描述符，它们分别是：⿰（左右结构，如

朋）⿱（上下结构，如志）⿲（左中右结构，如街）⿳（上中下结构，如意）⿴（全包围结构，如回）⿵（上三包围结构，如同）⿶（下三包围结构，如函）⿷（左三包围结构，如匠）⿸（左上包围结构，如居）⿹（右上包围结构，如忒）⿺（左下包围结构，如起）⿻（嵌套结构，如市）。

仍以鷟这个字为例，使用文字描述符即可表示为：⿱族鸟。这种方式比「左X右Y」或「上X下Y」更方便。使用这种方式描述一个生僻字时，需要使用括号括起来，避免与正文字混淆。

10.10.3 图片

生僻字图的制作见6.4节。生僻字图插入到段落内，见10.4.5一节。

10.10.4 外挂字库

制作生僻字图的方法虽然可行，不过当生僻字数量达到几十上百个时，就非常费时费力。所以我一般采用内嵌字体的方式去显示生僻字。具体的做法是通过使用Sigil的搜索模板来查找文中的生僻字，添加生僻字样式或者使用fallback方式附在正文字体样式后面。

生僻字字体采用开心宋体，这个是用来给正文内嵌字体的。字体先安装，然后重启浏览器，就可以完整显示所有字体。同样的在Sigil的书籍视图，也可以显示扩展区的字体，但是在阅读器中会不显示（可能是豆腐块□或者一个全角字符的空白，原因是移动设备上没有安装开心宋体），所以还需要子集开心字体到ePub中。而只要阅读器支持`@font-face`规则，就可以读取我们内嵌的子集字体。

以下是操作步骤：

1. 从国学大师网站上使用「部件查字」功能查找生僻字，复制生僻字粘贴到文中；
2. 运行搜索模板，给生僻字添加HTML结构；
3. CSS文件中增加生僻字和外挂字体样式；
4. 完成后在阅读器中预览效果。

如果我想查找一本书中的所有扩展A区、B区、C区、D区、E区的生僻字，那么我先在网络上找到这些扩展区的Unicode编码（见4.6节），然后转成Sigil支持的正则语法（例如汉字的正则通常是`[\u4e00-\u9fa5]`，但是在Sigil中则要写成`[\x{4e00}-\x{9fa5}]`，这是因为Sigil采用了16进制的Unicode编码传输方案）。那么正则表达式的搜索模板就可以这么写：

基本区补充	([\x{9fa6}-\x{9fcb}])	\1
扩展A区	([\x{3400}-\x{4db5}])	\1
扩展B区	([\x{20000}-\x{2a6d6}])	\1
扩展C区	([\x{2a700}-\x{2b734}])	\1
扩展D区	([\x{2b740}-\x{2b81d}])	\1
扩展E区	([\x{2b820}-\x{2cea1}])	\1

图10-22　生僻字样式搜索模板

Unicode汉字编码已经更新到F区了，如果有F区的生僻字需要显示，那么参考上一条的正则进行添加。

如果生僻字是连着的，还需要清除冗余代码：

```
查找：<span class="rarefont">([\x{9fa6}-\x{9fcb}]|[\x{3400}-
\x{4db5}]|[\x{20000}-\x{2a6d6}]|[\x{2a700}-\x{2b734}]|[\x{2b740}-
\x{2b81d}]|[\x{2b820}-\x{2cea1}])</span><span
class="rarefont">([\x{9fa6}-\x{9fcb}]|[\x{3400}-\x{4db5}]|
[\x{20000}-\x{2a6d6}]|[\x{2a700}-\x{2b734}]|[\x{2b740}-\x{2b81d}]|
[\x{2b820}-\x{2cea1}])</span>
替换：<span class="rarefont">\1\2</span>
```

上面的清除冗余正则也可以添加到搜索模板中，放在扩展E区条目下面。

把下面的CSS代码添加到生僻字内容文档所引用的CSS文件中：

```css
@font-face {
    font-family: "KaiXinSong";
    src: url("../Fonts/KaiXinSong.ttf");
}
span.rarefont {
    font-family: "KaiXinSong";
} /*生僻字字体*/
```

HTML代码：

```html
<p>器服物佩好无疆。曰天子一月休。秋癸亥，天子觞重<span class="rarefont">𨛱</span>之人<span class="rarefont">䐴
𪕋</span>，乃赐之黄金之罂二九，银乌一只，贝带五十，朱七百裹，<span class="rarefont">筍</span>箭桂姜百<span class="rarefont">崀</span>，丝<span class="rarefont">䋇</span>雕官，<span class="rarefont">䐴𪕋</span>乃膜拜而受。</p>
```

然后参考8.4节，子集生僻字字体。完成后导入到多看阅读器中，找到生僻字出现的章节，显示如下：

7:07 65%

卷四

三百。牛羊五千。秋麦千车。秋麦，禾也。膜稷三十车，稷，粟也。膜，未闻。天子使柏夭受之。好献枝斯之英四十，精者为英。[illegible][illegible][illegible][illegible]珌佩百只，琅玕四十，[illegible][illegible]十箧，疑此纻葛之属。天子使造父受之□，乃赐之银木[illegible]采，黄金之罂二九，贝带四十，朱三百裹，桂姜百[illegible]，[illegible]奴乃膜拜而受。

乙亥，天子南征阳纡之东尾，尾，山后也。乃遂绝[illegible]䇉之谷。已至于[illegible]瑀，河之水北阿，爰有[illegible]溲之□。河伯之孙西有渠搜国，疑[illegible]渠字。事皇天子之山，有模堇，其叶是食明后，模堇，木名。后，君也。堇音谨。天子嘉之，赐以佩玉一只，柏夭再拜稽首。

癸丑，天子东征，柏夭送天子至于[illegible]人。[illegible]伯絜觞天子于澡泽之上，[illegible]多之汭，汭，水涯。河水之所南还。还，回也。音旋。曰天子五日休于澡泽之上，以待六师之人。

戊午，天子东征，顾命柏夭归于其邦。天子曰："河宗正也。"柏夭再拜稽首。辞去也。

07:07 32 / 51



图10-23 生僻字内嵌效果

> 上图中的□，原文即是如此。

上面说的是使用现有的大字库，通过子集字体的方式解决生僻字显示问题。但两种字体放在一起，明显有风格不搭的情况。如果追求完美，显然是不能接受的。所以还需要使用现有的正文字体部件，然后拼成一个生僻字。这样就能解决风格不一的问题了。

10.10.5 自造字

自造字的缺点是对技术操作、设计能力要求比较高，所以最好是只造一些结构不复杂的字。如果强行为了显示一个生僻字而造字，字的部件会不协调，没有设计感，还不如使用现成的字库。关于造字的教程，可参看阁主的电子书外挂生僻字教程（#1，#2，#3）。

造字之前，先看下这个字的Unicode码位，如果有就用，没有就自编号。Unicode中预留了平面可以存储自造字的码位，从F0000开始。码位的规则是从1开始到9，然后再从A开始到F。依此循环。比如：

```
F0000、F0001、F0002、F0003、F0004、F0005、F0006、F0007、F0008、F0009、
F000A、F000B、F000C、F000D、F000E、F000F、F0010、F0011……
```

知道了规则之后，开始造大批量的生僻字就好办了。这里以「□舛犬」为例（该字出自2011年上海书画出版社《安持人物琐忆》第141页）。

准备

多看阅读中正文字体为方正书宋，那么我们就是用书宋的部件作为素材。实际上在多看阅读中的书宋字体，与方正字库中的书宋要粗一些，造字基础字体可以用多看阅读自带的书宋字体。以小米8手机为例，打开「文件管理」应用，顶部菜单项选择「手机」，依次找到下面文件夹：

```
...>DuoKan>Resource>Font>方正书宋_GBK.ttf
```

字体编辑软件FontLab。下载之后，解压文件，双击运行「FLS5WinFull.exe」，执行安装。安装完成后，运行「FLS_RegPatch.exe」破解。

确认码位

在国学大师网部件查字页面中，输入所造字部件查询。如果有，则使用当前码位，如：

汉字：泉

拼音：

部首：水　四角码：　仓颉：jre

86五笔：diu　98五笔：DIU　郑码：EDJK

统一码B：23CC2　总笔画数：9　笔顺：122512534

图10-26　汉字信息

「统一码B：23CC2」中的「23CC2」就是该字在Unicode中的码位。我们要造的「□舛犬」没有收录到Unicode中，所以需要我们自己确定一

个码位。如果你的电脑中已经安装过一些全字库字体，它的预留码位起始部分可能被占用了。所以我们就不从F0000开始，而从F1100作为第一个码位。

查找共用部件

我们现在要造的字属于上下结构，那么就应该找同样是上下结构的汉字。「舛」在上，「犬」在下。首先现在部件查字中搜索「舛」，结果如下：



图10-27　「舛」部件所组成的字（部分）

由于方正书宋是GBK字库，只包含了Unicode中的基本区。所以在选用时，只能选择灰底的汉字。这里选择「桀」，记下它的码位「6840」。然后是「犬」，这里选择「臭」，记下它的码位「81ED」。

造字

启动FontLab，使用快捷键「Ctrl＋N」新建一个项目，或者依次点击「File→New」。此时窗口如下：

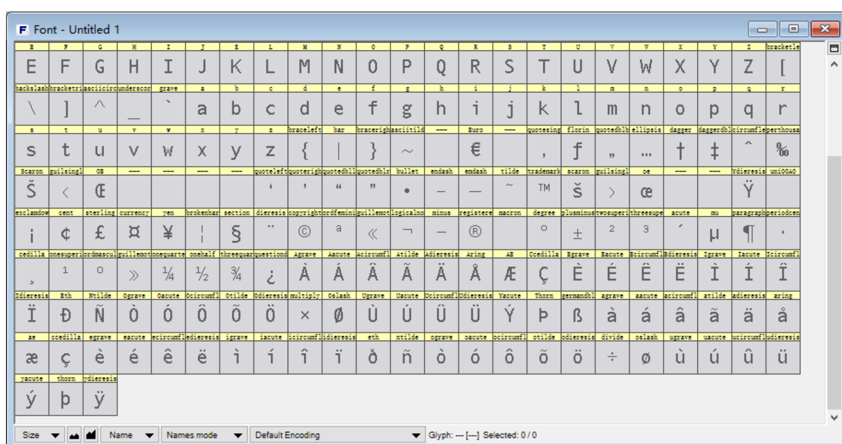


图10-28　FontLab新建项目

使用快捷键「Ctrl＋O」打开方正书宋字体文件，在「方正书宋」字体窗

口中使用快捷键「Ctrl＋F」，在弹出窗口中输入「6840」回车，如下：

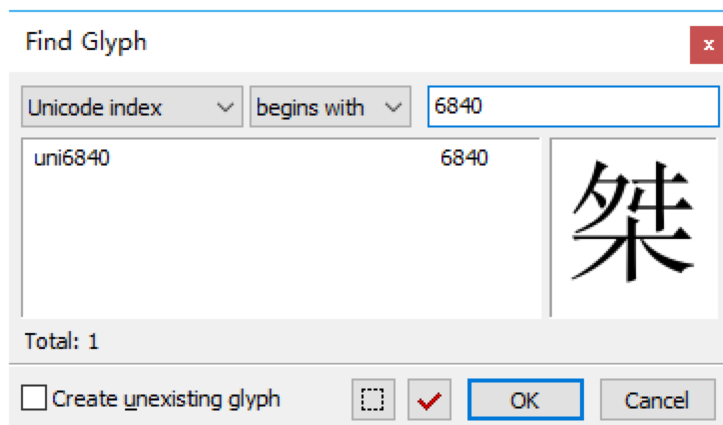


图10-29　「桀」

注意，搜索条件限制为「Unicode　index」，点击「OK」按钮跳转到该字形所在位置：

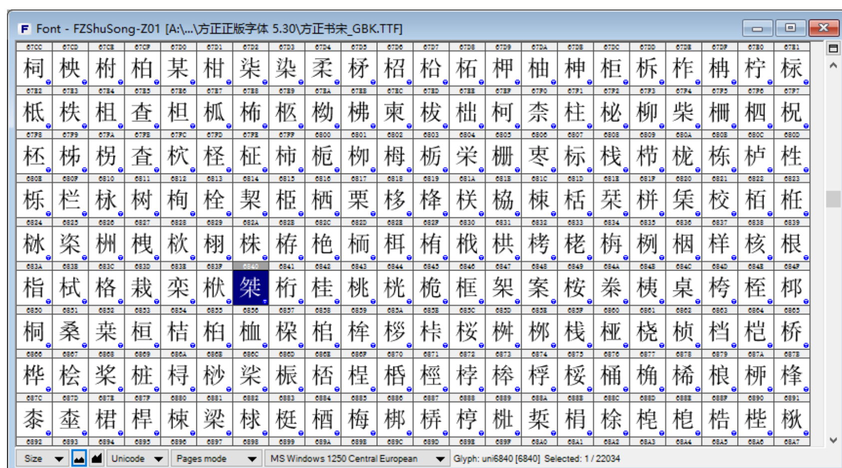


图10-30　「桀」字形

此时被选中的字形底色为蓝色，按快捷键「Ctrl＋C」复制该字形，切换到之前新建的项目窗口「Untitle-1」中，不要选中任何字符，使用快捷键「Ctrl＋V」粘贴刚才复制的字形。

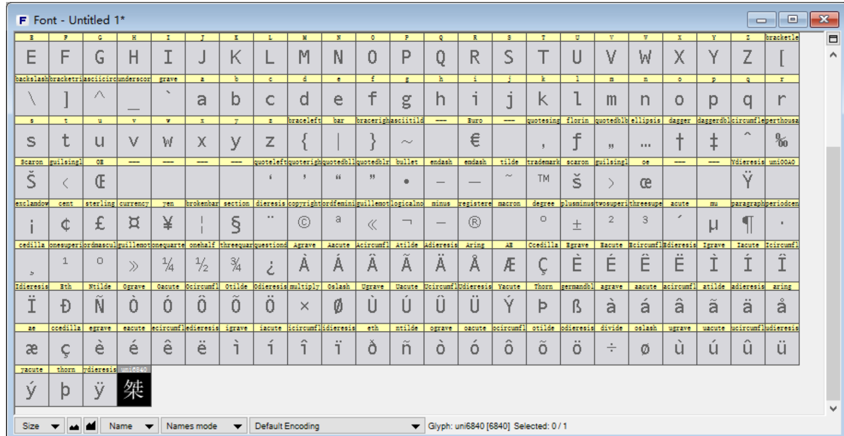


图10-31　「桀」字形

这样就完成字符的借用，选中「Untitle-1」窗口中的「桀」，点击鼠标右键或使用快捷键「Ctrl＋\」，选择「Rename...」选项：

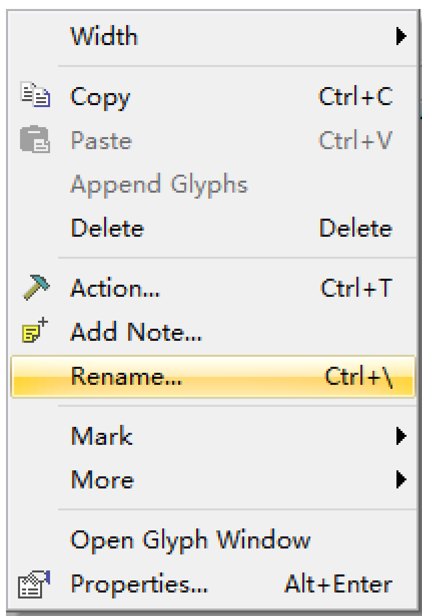


图10-32　修改码位

在弹出窗口中修改这个字形的码位：

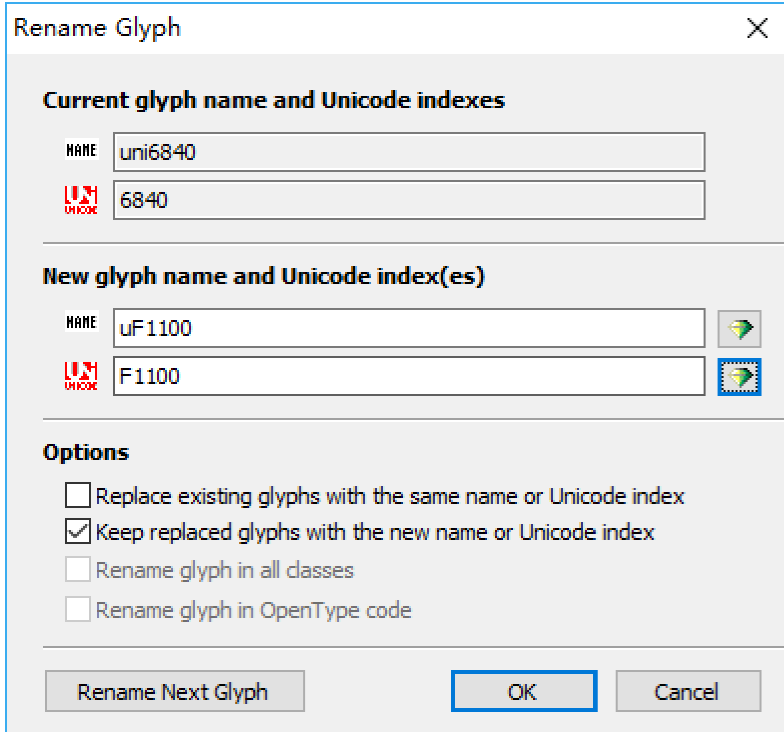


图10-33　重命名码位

字符名称中输入「uF1100」，Unicode编码中输入「F1100」。然后点击「OK」按钮即可。

现在我们需要编辑这个字形，删掉我们不需要的部件「木」。在「Untitle-1」窗口中双击「桀」，进入编辑窗口：

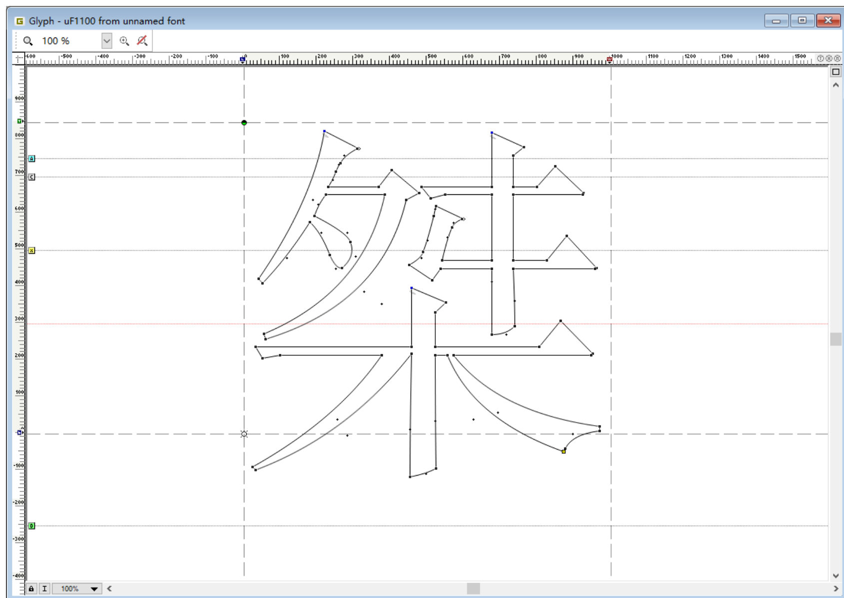


图10-34　字体编辑窗口

双击「木」字的一条线段，会自动选中该线段所在的封闭区域。被选中的结构会变成红色，按「Delete」键即可删除。

现在回到「方正书宋」字体窗口中，搜索「臭」字，双击该字形进入到字体编辑窗口中，双击「犬」，选中后使用快捷键「Ctrl+C」复制。切换到「桀」字的编辑窗口，「Ctrl+V」粘贴「犬」部件，完成后如下：

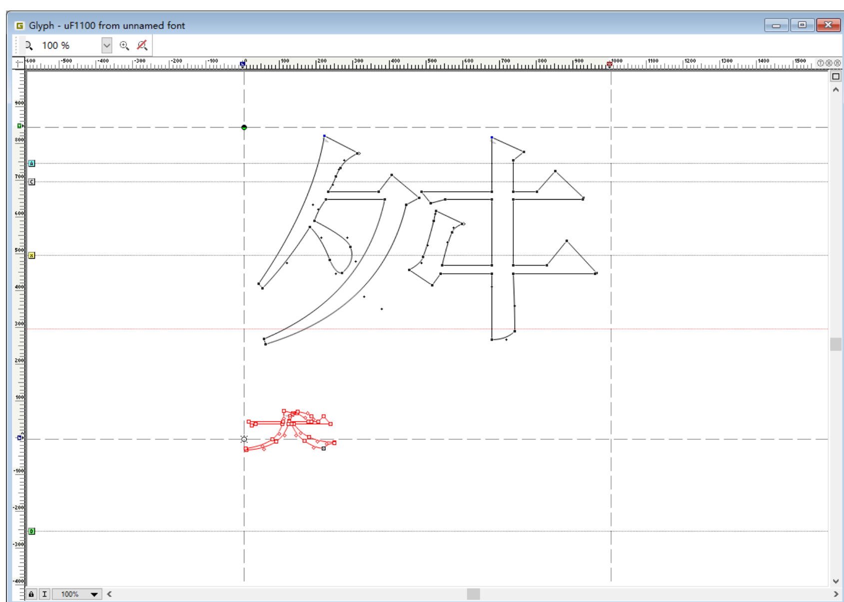


图10-35　字体编辑窗口

FontLab更多的操作方式见阁主写的教程。

可以看到这两个部件大小并不一致，点击顶部快捷按钮，红色箭头所指的：



图10-36　放大部件

此时字体编辑显示如下：

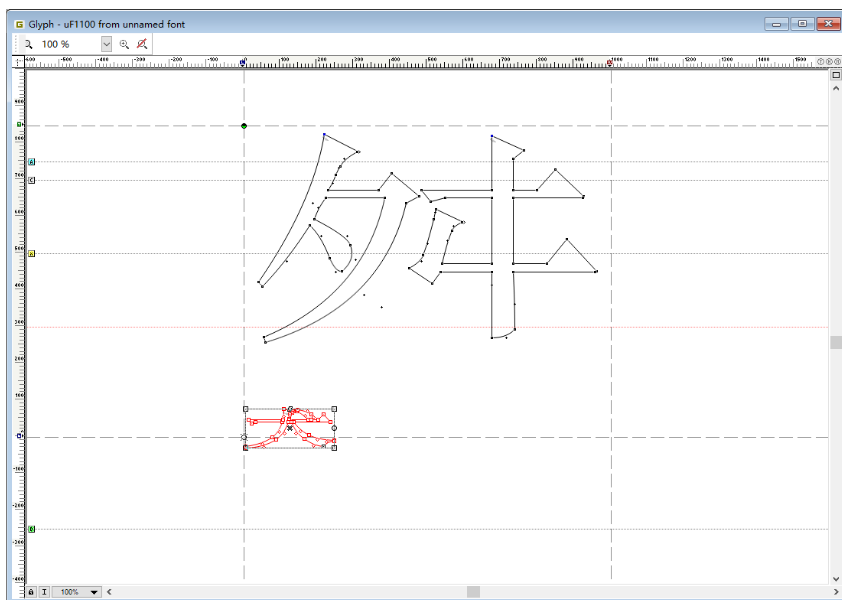


图10-36　放大部件

按住「Shift」键，鼠标移动到4个角中的任意一个，拉伸放大到合适大小。再使用键盘上的箭头键移动该部件，调整如下：

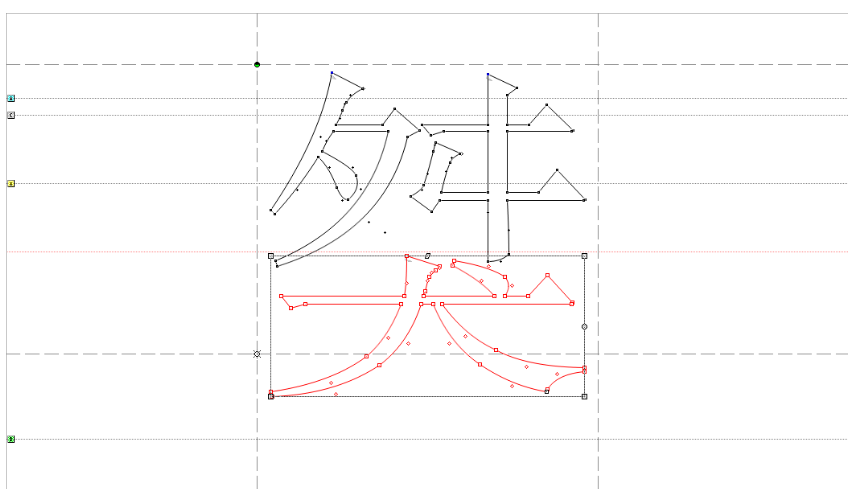


图10-37　调整部件

箭头＋Shift键可以一次移动较大的位置，单独使用箭头键则可以进行微调。

至此，就完成一个汉字的拼造。使用快捷键「Ctrl＋Alt＋F」调出字体信息窗口。

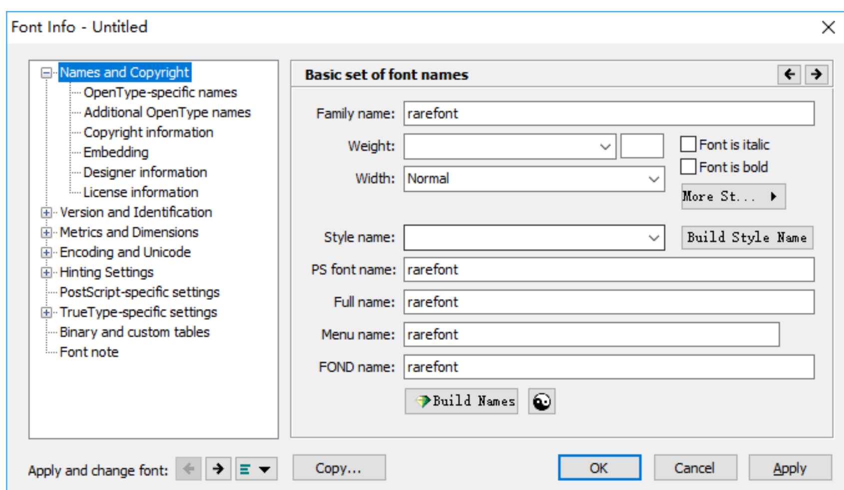


图10-38　字体信息编辑

点击左侧的「Names and Copyright」条目，在「Family name」中填写字体名称，这里使用「rarefont」；点击「Bulid Names」按钮。然后点击「Apply」应用该信息。

切换到「Copyright information」条目，填写字体设计主体（人或公司）。在「Created by」中填写好之后，点击「Bulid Copyright and

Trademak …」，然后点击「Apply」应用该信息。

在「Embedding」条目中，设置字体嵌入性的权限为「Everthing is allowed (installable mode)」，然后点击「Apply」应用该信息。

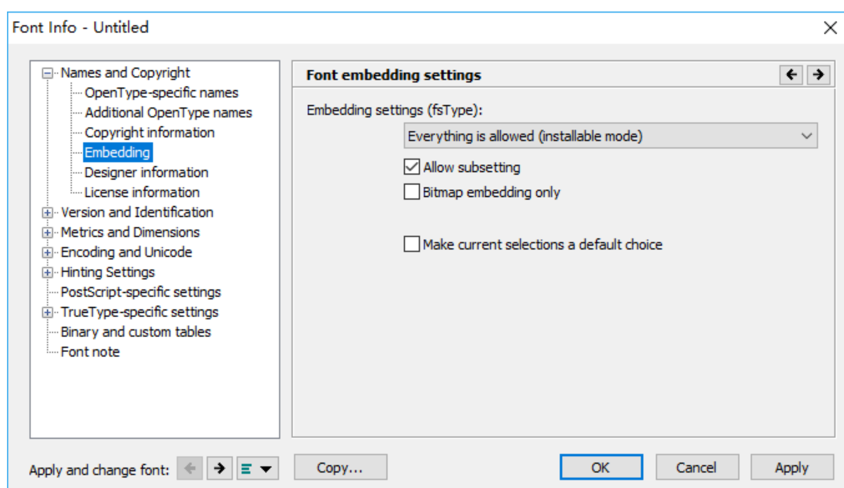


图10-39　字体嵌入性

完成后点击「OK」按钮退出。

使用快捷键「Ctrl+S」或依次点击「File→Save」保存，选择一个文件夹，然后点击「保存」即可。

接着创建ttf格式字体文件，使用快捷键「Ctrl+Alt+G」，在弹出窗口中点击「保存」即可。此时我们就获得了一个字体文件，下面需要做的就是用css来使用这个字体了。

显示测试

创建一个空白ePub文件，添加*rarefont.ttf*到ePub中，HTML代码如下：

```
<body>
   <p><span class="rarefont">&#xF1100;</span>是一个不存在的字，创造它是为了解决生僻字在电子书中的显示问题。</p>
</body>
```

以HTML支持的实体方式输入该字。创建一个css样式表，代码如下：

```
@font-face {
    font-family: "rarefont";
    src: url("../Fonts/rarefont.ttf");
}
.rarefont {
```

```
    font-family: "rarefont";
}
p {
    line-height: 1.35em;
    font-family: "DK-SONGTI";
}
```

然后链接样式表就行了。保存这个ePub文件，将该ePub传输到手机中，使用多看阅读打开：

> [illegible]是一个不存在的字，创造它是为了解决生僻字在电子书中的显示问题。

以上就是生僻字的制作。

10.10.6　图片字做字体

还有一个问题，那就是字符是一种图片字。比如：



图10-40　金文「龙」

遇到这种情况，通常是把背景删掉，做成透明图，然后插入到书中。但现在我们有别的方式，可以把图片转成字体。下面是详细教程。

下载ScanFont，解压后，运行「SFont5.exe」。初次使用该软件时，需要填写用户名，随便输入一串字符就可以了。

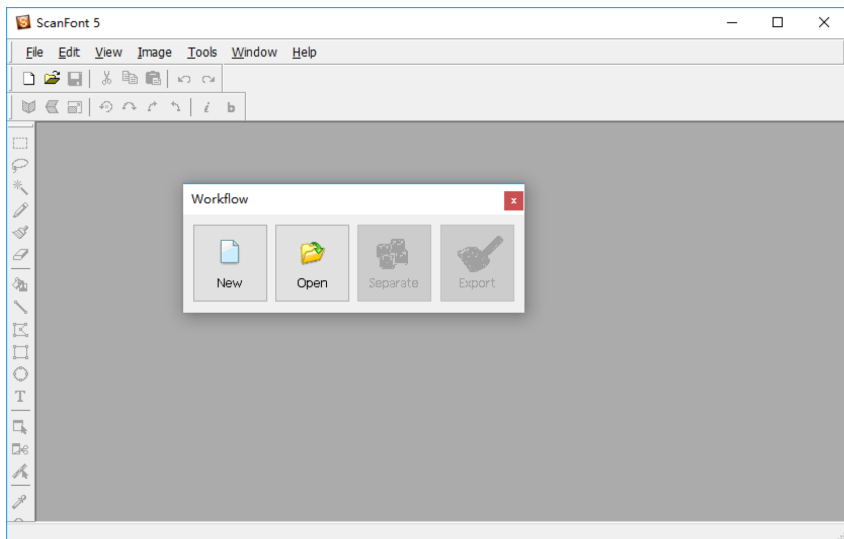


图10-41　ScanFont界面

依次点击「Tools→Options...」或者使用快捷键「F10」，在弹出窗口中，点击左侧「Outline Font Editor」条，勾选「Export outline font into VFB file」，点击「OK」按钮保存并退出。这个操作是为了保存时把项目存储为vfb格式文件。

依次点击「View→Toolbars→Workflow」也可隐藏Workflow窗口。

点击「Open」按钮，打开上文中的金文「龙」图。

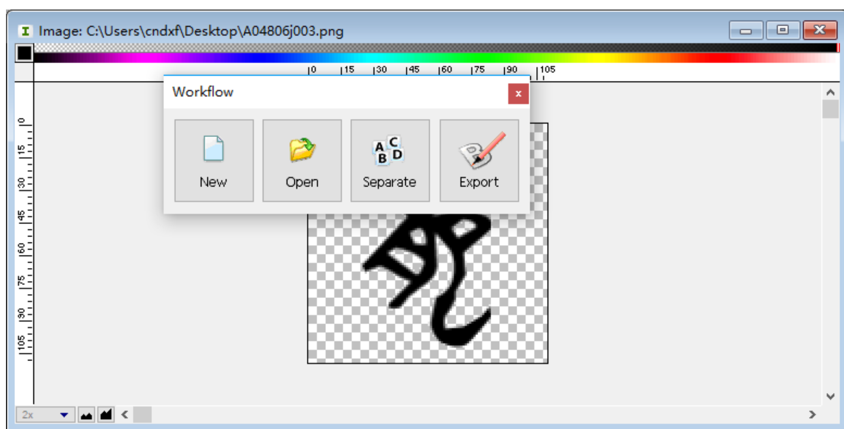


图10-42　打开「龙」图

点击「Separate」按钮，切分。弹出如下窗口：

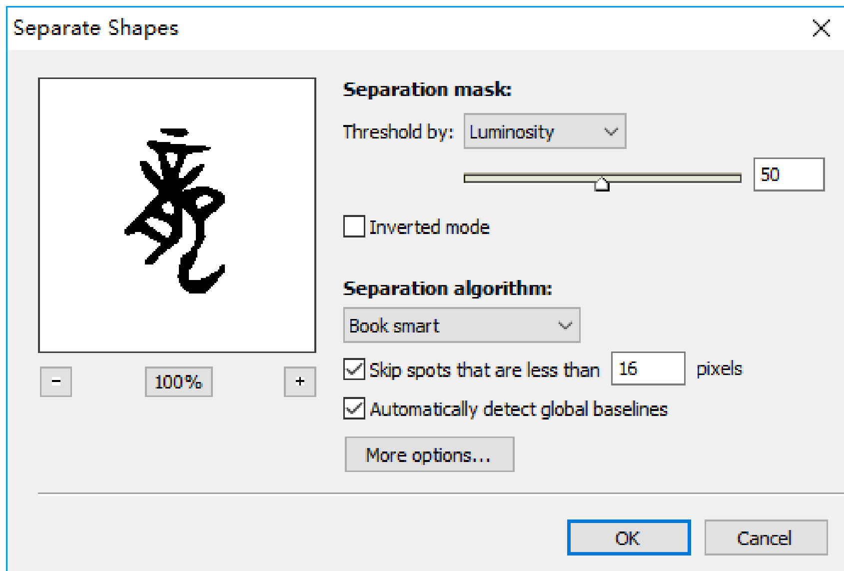


图10-43　设置字形

「Luminosity」指的是亮度，默认是50，如果线条显得纤细，可以再调整。然后点击「OK」按钮退出。

点击「Export」按钮，分配码位。

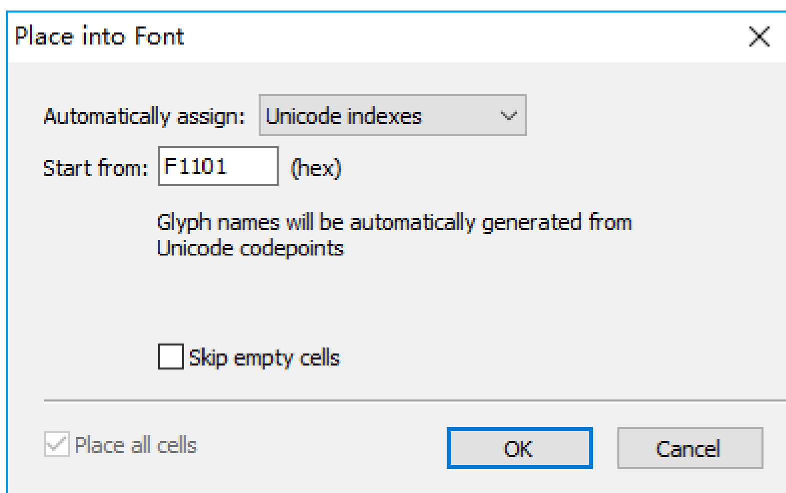


图10-44　分配码位

在「Start　From」中输入码位，点击「OK」按钮退出。然后弹出窗口如下：

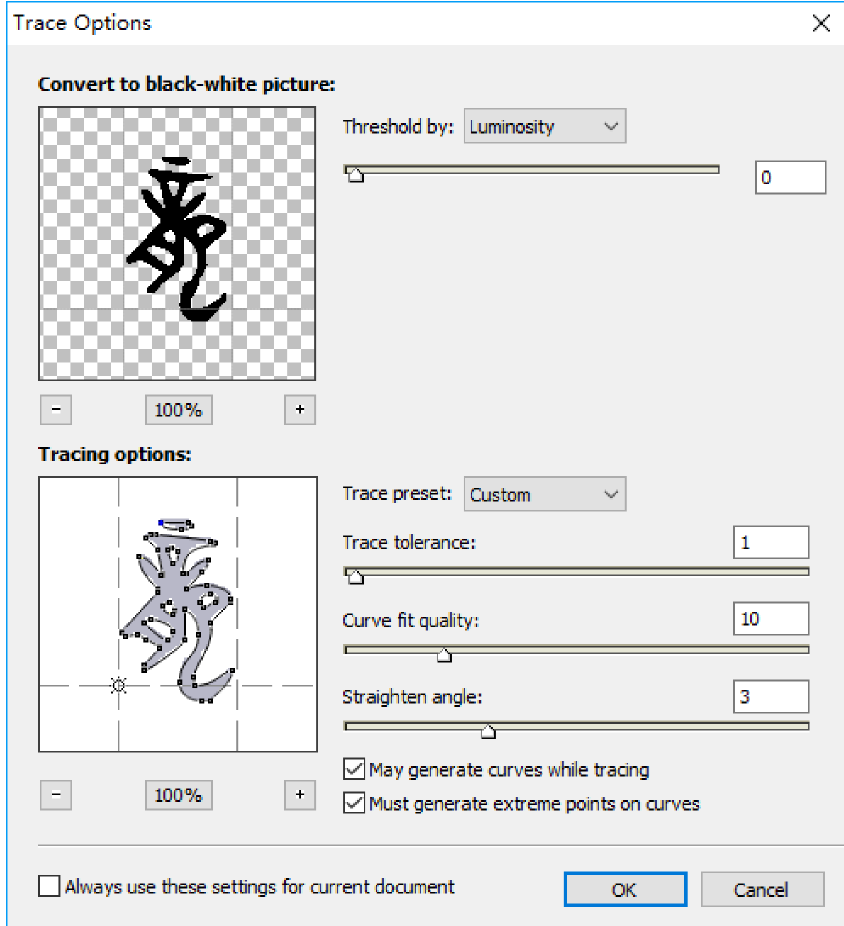


图10-45 字体配置

不需要做任何操作，直接点击「OK」按钮退出。接着会弹出一个保存文件的窗口，给文件命名之后，点击「保存」按钮即可。这里文件命名为「F1101.vfb」。

启动FontLab软件，打开生僻字项目文件，这里是「rarefont.vfb」。然后打开「F1101.vfb」文件，选择字形路径，复制到新的码位中，调整如下：

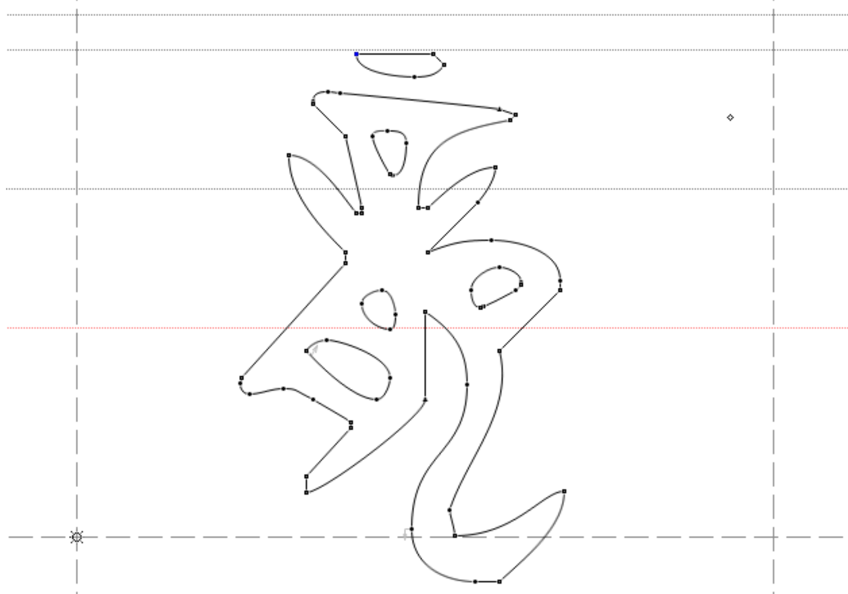


图10-46　「龙」字体

保存之后，创建字体。参考上文的操作，制作ePub。在多看阅读中显示如下：

> 龙 是传说中的一种长形、有鳞、有角的神异动物，能走，能飞，能游泳，能兴云作雨。

以上就是自造字的全部教程。

10.11 多看扩展效果

本节内容主要讲下在多看阅读器中实现的特殊效果，比如波浪下划线、全屏展开、贴边、正文字体自定义且保持排版效果等。多看阅读器是个非常棒的阅读器，强烈推荐您使用。

10.11.1 贴边

在前文中提过贴边图的实现是使用负边距的方法，紧贴屏幕边缘。实际上多看阅读器扩展这个功能。CSS代码：

```
duokan-bleed: lefttopright | top | left | right | bottom | lefttop | topright | leftright;
```

HTML代码示例：

```
<div class="title-bleedimg">
  <img alt="chapter1" src="../Images/chapter1.png"/>
</div>
```

给类title-bleedimg应用贴边图`lefttopright`样式。在多看阅读器和预览视图中的显示效果如下：



第一章

濮阳之血

日昏黄，暮苍茫。彤云如絮，掠过黯淡的苍穹，将天空划出一道血口，染红垂天云翼，一只淌血的孤雁，盘旋在疮痍满目的大地之上，悲凉静肃地凝视着即将颓倾的濮阳楼堞。

名城遭戮，天地寂寂。

濮阳之血

图10-48　Sigil中的贴边图

日昏黄，暮苍茫。彤云如絮，掠过黯淡的苍穹，将天空划出一道血口，染红垂天云翼，一只淌血的孤雁，盘旋在疮痍满目的大地之上，悲凉静肃地凝视着即将颓倾的濮阳楼堞。

图10-47　贴边图阅读界面

10.11.2　全屏展开

全屏展开图是指当打开一个新页面时只显示一半内容，点击展开图标再显示另外一半内容。效果如下：

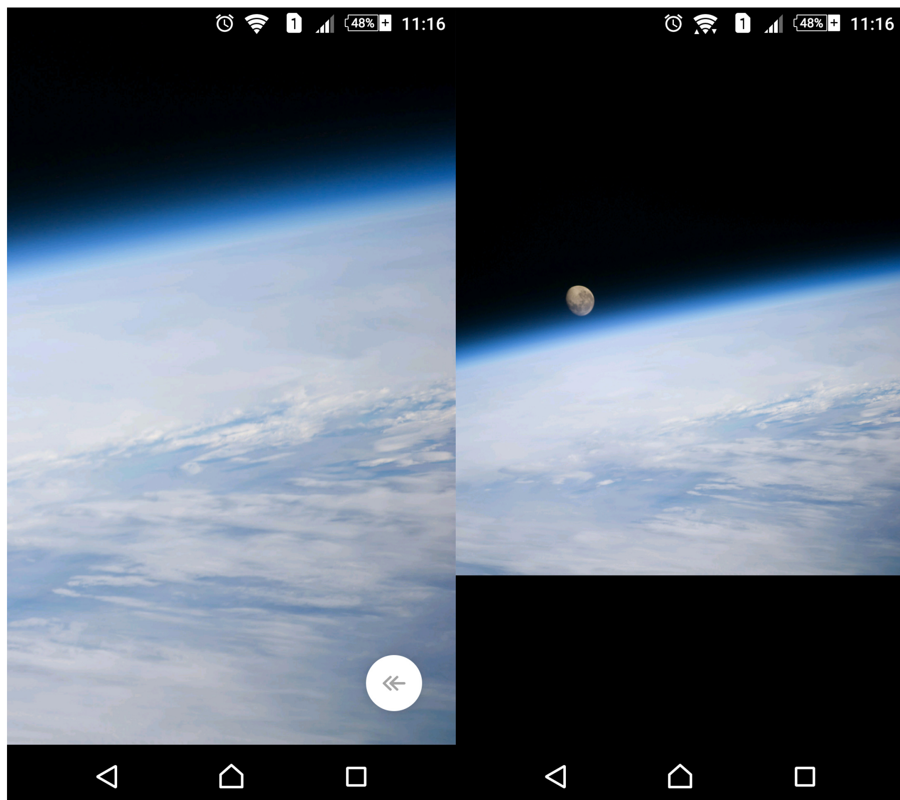


图10-49　全屏展开图

显示右隐藏左

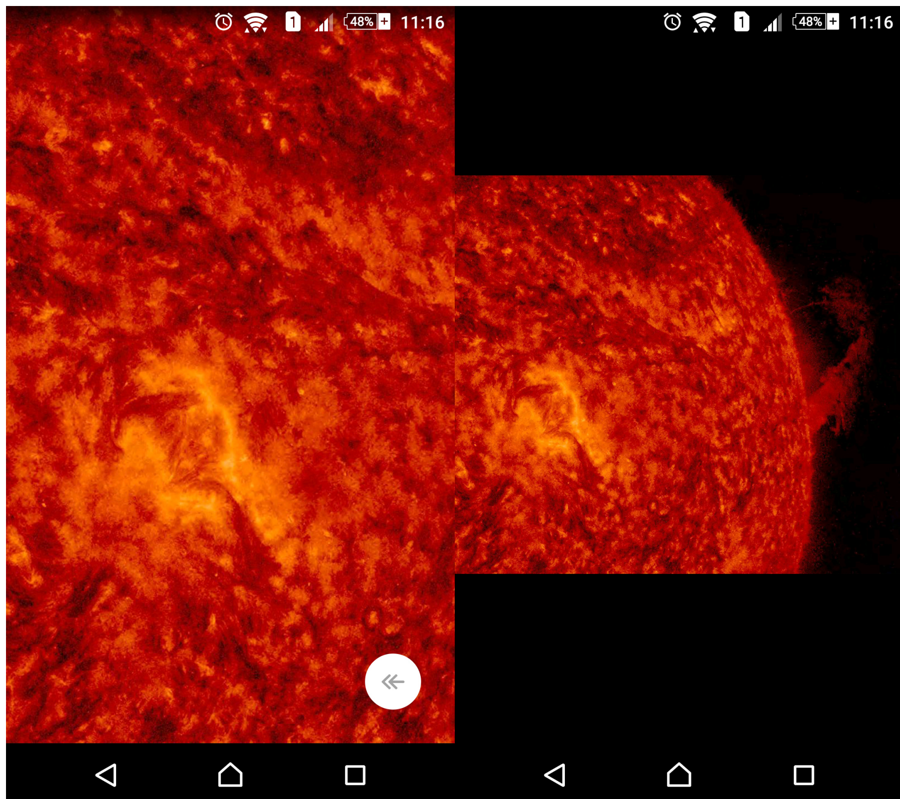


图10-50　全屏展开图

显示左隐藏右

制作要求如下：

1.xhtml页代码（一个xhtml页面只能添加一张图片）：

```
<!--speed-right.xhtml-->
<body>
   <div class="speed"><img alt="" class="speed" src="../
Images/speed-1.jpg"/></div>
</body>
<!--speed-left.xhtml-->
<body>
   <div class="speed"><img alt="" class="speed" src="../
Images/speed-2.jpg"/></div>
</body>
```

2.css代码：

```
div.speed {
    text-align: center;
    margin: 0;
```

```
    padding: 0;
}
img.speed {
    width: 100%;
}
```

3.图片要求

比例关系（18w:16h），如以下尺寸：1440px*1280px；2880px*2560px；2160px*1920px。

4.opf要求

```
<spine toc="ncx">
  ...
  <itemref idref="speed-right.xhtml" properties="duokan-page-fullscreen-spread-right"/>
  <itemref idref="speed-left.xhtml" properties="duokan-page-fullscreen-spread-left"/>
  ...
</spine>
```

10.11.3　波浪下划线

下划线一般使用text-decoration属性实现，也可以给行内元素添加border来实现不同风格的下划线，比如点线、虚线、实线、双线。

专名号

专名号，又称私名号，用于标明人名、地名、朝代名、种族名、国名、机构名等专名所使用的符号。它的使用方法是横排时划在专名之下，竖排时则划在专名左旁。

专名号在近年的中文出版物中已十分罕用，但台湾和香港中文教科书仍会教授使用，电视或电影中文字幕中仍偶见保留；中国大陆则只用于古籍、文史哲作品，与浪线书名号配合使用。

　　黃帝者，少典之子，姓公孫，名曰軒轅。生而神靈，弱而能言，幼而徇齊，長而敦敏，成而聰明。

　　軒轅之時，神農氏世衰，諸侯相侵伐，暴虐百姓，而神農氏弗能征。於是軒轅乃習用干戈以征不享，諸侯咸來賓從。而蚩尤最爲暴，莫能伐。

CSS代码：

```
span.text-solid {
    border-bottom: 1px solid #333;
    padding-bottom: 1px;
}
```

上面的代码与`text-decoration: underline;`效果类似。

书名号

中国大陆书名号的基本形式有两种，即双书名号（《　》）和单书名号（〈　〉），例如马克思《摩尔根〈古代社会〉一书摘要》，外双内单。浪线书名号，现在仅在古籍配合专名号一起使用。

　　史記原名太史公書，司馬遷撰。司馬遷字子長，漢左馮翊夏陽（今陝西韓城縣）人，生於漢景帝中五年（公元前一四五）或者更後一些。他的父親司馬談，熟悉史事，懂天文地理。漢武帝建元（公元前一四〇——一三五）初年，做了太史令（史記中稱爲太史公）。他早就有意論載「天下之史文」，但始終沒有如願。

HTML代码：

```
<p class="bodycontent"><span class="text-wavy">史記</span>原名<span class="text-wavy">太史公書</span>，<span class="text-solid">司馬遷</span>撰。...初年，做了太史令（<span class="text-wavy">史記</span>中稱爲太史公）。他早就有意論載「天下之史文」，但始終沒有如願。</p>
```

CSS代码：

```
span.text-wavy {
    /* Firefox 写法*/
    text-decoration-style: wavy;
    -moz-text-decoration-style: wavy;
    /*使用图片实现波浪线*/
    border-bottom: 4px solid transparent;
    border-image: url("../Images/bolang.png")10 7 7 7 round;
    /*多看补丁属性*/
    text-decoration: duokan-wavyline;
    duokan-text-decoration-width: 1px;
    duokan-text-decoration-color: #333;
    padding-bottom: 1px;
}
```

浪形下划线目前只有火狐浏览器可以正确显示。第2种方法是使用图片绘制，使用`border-image`；第3种是基于阅读器的扩展属性。

10.11.4　正文字体自定义

在多看阅读器中阅读自制书一般建议设置如下：

> 在设置中将排版设为「原版」（多看2.x版本）或「无」（多看3.x版本以上），背景为预设背景（不要自定义背景和字体颜色，以免整体配色出问题）；字体设置为「默认」（使用书中指定字体），字体大小为默认大小（一般手机上为+3，平板上为+2——即减小字体到最小值后，点击增大按钮的次数）。

完成以上设置之后，再打开一个TXT文本，就非常不适合阅读。（下面的图片特意调成横屏状态，主要是为了图片的显示效果。）

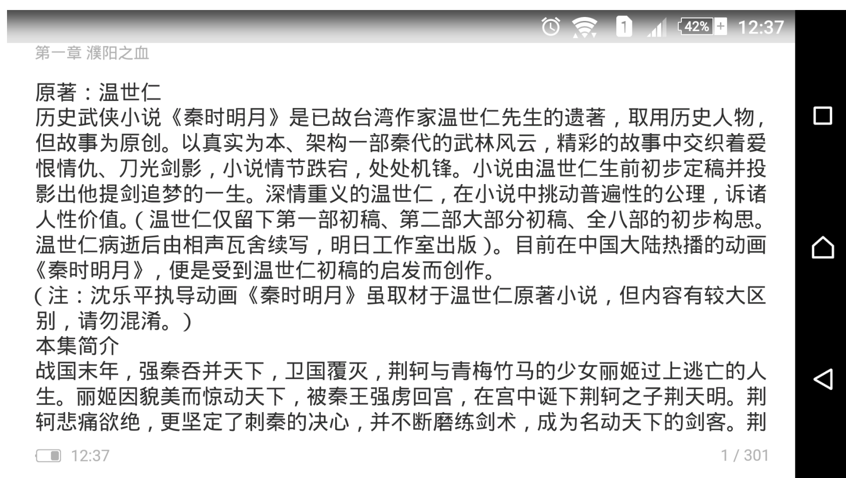


图10-51　TXT文本的默认效果

可以看到TXT默认是没有行间距、段间距的，正文是系统字体，为了舒适的阅读，你可能会设置「排版」效果，从字体包中选择一个字体作为正文字体。这是当你阅读TXT时做的设置，如果你打开一本ePub电子书，则又需要按照本节开头中推荐的设置，去设置你的ePub。

所以为了避免这种情况的出现，本节介绍一种方法，即可以自定义正文字体，又能保证排版效果，同时也不影响TXT的阅读。

打开ePub中的opf文件，添加多看的识别符（duokan-book-id）和正文字体元数据条目：

```
<... unique-identifier="duokan-book-id" ...>
  <metadata ...>
    ...
    <dc:identifier id="duokan-book-
id" opf:scheme="UUID">...</dc:identifier>
    ...
```

```
        <meta content="DK-SONGTI" name="duokan-body-font" />
    </metadata>
```

> unique-identifier的值必须和dc:identifier元素的ID属性匹配。

样式表中如果有设置text-indent的样式，需要再使用多看的自定义缩进属性duokan-text-indent重新声明。例如单独声明text-indent: 0em;没有用，还需要再声明如下：

```
duokan-text-indent: 0em;
```

其他设置了缩进值的text-indent一样需要再声明duokan-text-indent。

如此，方可以从多看的字体包中选择你喜欢的字体，而排版效果不变。添加了duokan-book-id这个内容，就相当于伪·正版书。

该方法由@肖奈、@云轩阁主、@阿狗共同发现，感谢！

10.11.5 使用多看字体

在初次使用多看阅读器时，可以下载多看字体包，即包含下面描述的字体。

多看官方客户端可用字体列表（示例）：

表28 多看阅读器字体CSS写法

CSS写法	效果
font-family: "DK-SONGTI";	使用多看系统自带宋体。
font-family: "DK-FANGSONG";	使用多看系统自带仿宋。
font-family: "DK-KAITI";	使用多看系统自带楷体。
font-family: "DK-HEITI";	使用多看系统自带黑体。
font-family: "DK-XIAOBIAOSONG";	使用多看系统自带小标宋。
font-family: "方正悠宋＋ 简 507R";	使用多看系统自带方正悠宋。
font-family: "DK-XIHEITI";	使用多看系统自带细黑体。
font-family: "DK-SERIF";	使用多看系统自带衬线西文字体。 Use the DuoKan system comes with the serif font.
font-family: "DK-CODE";	使用多看系统自带等宽西文字体。 Use the DuoKan system comes with the monospaced font.
font-family: "DK-SYMBOL";	使用多看系统自带符号字体（不常见符号，如音标等）。 shǐ yòng duō kàn xì tǒng zì dài fú hào zì tǐ（bú cháng jiàn fú hào ，rú yīn biāo děng）。

首先在CSS文件中增加下面的样式

```
p.usekaiti {
```

```
  font-family: "DK-KAITI";
}
```

然后在xhtml文件中使用

```
...
<p class="usekaiti">这段文字使用楷体显示</p>
...
```

这样就可以了。

10.11.6　幻灯片（漫画）

这个教程来自云轩阁主写的怎样制作一本多看漫画书，感谢！

多看书城的书中，有一部分是漫画书，结构功能和普通的图书有一点点不一样：

1. 每一页图片居中显示；
2. 每一小格图片格可以双击放大，进入幻灯片模式；
3. 单击屏幕中央，系统菜单也会出现一个进入幻灯片模式的图标；
4. 进入幻灯片模式之后，可以左右滑动一直往下看；
5. 书架上漫画书的封面会有「漫画」标签做标识。

其实这类漫画书也很容易做，不需要你在每个页面加工什么特殊代码，只需要改epub的核心——OPF文件就行。

多看阅读器中实现垂直居中需要在OPF文件中添加扩展属性，与全屏封面类似。要求：

HTML：网页中有且只有1张图片，即使添加了文字也不会显示；

```
<body>
  <p><img alt="slide" src="../Images/slide.jpg"/></p>
</body>
```

OPF：添加扩展属性，如：

```
<spine toc="ncx">
  ...
  <itemref idref="color-page.xhtml" properties="duokan-page-fitwindow"/>
  ...
</spine>
```

图片不能遮盖到的部分会根据阅读器设置的阅读背景显示，就算你在页

面中定义了页面背景也没用，比如颜色、图案。

然后是使用坐标定位每一格，阁主在教程里写的是用Ps来查看坐标，我没看懂，所以我用了别的方法。

首先下载马克鳗，点击下载时提示你需要先安装Adobe AIR，这个程序安装完以后，再安装马克鳗。安装完以后，运行马克鳗。打开做好的图片，使用软件底部正数第2个工具——「坐标/范围标记」，标记图片的左上角、右下角的坐标（如果想要精确标记，可以按住Ctrl键，同时滑动滚轮，放大图片）：



图10-52　图片四角坐标

其余5张图片也只需要标记左上角和右上角的坐标，全部标记完以后，记录坐标。然后做成下面这种格式：

```
...
</spine>
<duokan-comics-thread>
  <frame boundary="48 48 1247 832" idref="color-page.xhtml" />
  <frame boundary="48 1289 1247 2111" idref="color-page.xhtml" />
  <frame boundary="2528 1289 3791 2111" idref="color-page.xhtml" />
  <frame boundary="1288 1289 2487 1827" idref="color-page.xhtml" />
  <frame boundary="1288 48 2087 1247" idref="color-page.xhtml" />
  <frame boundary="2128 48 2937 1247" idref="color-page.xhtml" />
```

```
  <frame boundary="2978 48 3791 1247" idref="color-page.xhtml" />
</duokan-comics-thread>
<guide>
...
```

6条代码就表示有6张图。需要注意的是Sigil 0.89以上版本因为改动了架构，对OPF要求比较严，无法写入多看漫画代码了。所以要做多看漫画的话，使用解压缩软件，比如用Bandizip打开ePub文件，直接编辑OPF文件，把漫画代码添加进去，保存然后退出ePub文件。

你可以下载我制作的案例《咏花令》查看此ePub的实际效果，案例所使用的图片来自@摄影师蝈蝈小姐 - 咏花令。

10.11.7 多看书城漫画（旧）

这是一种与上一节不同的漫画格式，制作技术来源简书的 @米小刀 。现整理如下：

每个内容文档中一张图片：

```
<body>
  <p><img alt="picture" src="../Images/picture.jpg"/></p>
</body>
```

CSS设置如下：

```
body, p, img {
    margin: 0;
    padding: 0;
}
```

在opf文件修改并增加如下：

```
<package version="2.0" unique-identifier="duokan-book-id" xmlns="http://www.idpf.org/2007/opf">
    ...
    <meta content="standard" name="duokan-resource-quality" />
    <meta content="trial" name="duokan-book-contenttype" />
    <dc:identifier opf:scheme="UUID" id="duokan-book-id">...</dc:identifier>
...
```

`unique-identifier`属性添加`duokan-book-id`内容，并增加两行meta。从名

称可知，这仅为试验性质的技术，所以不保证该格式在以后的版本中仍可正常浏览。

做好以上工作后，使用WinRAR等压缩文件软件打开ePub文件，在「META-INF」文件夹下增加一个名为*duokan-extension.xml*的文件，xml文件内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<duokan-extension version="2.4.0">
  <display-options layout="vertical-comic"/>
  <writing-options>
    <option name="writing-mode">horizontal-tb</option>
    <option name="direction">ltr</option>
  </writing-options>
</duokan-extension>
```

注意：当使用Sigil打开已经添加了xml文件的ePub文件，如果修改了ePub文件，该xml文件会消失，需要重新添加。

制作好的ePub电子书，在导入到多看阅读器中，会在书架封面上出现「漫」图标，打开后会隐藏掉系统信息。此时翻页的话仍是左右翻页，如果想上下滚动的话，需要在设置里把「翻页动画」设置为「上下」。效果上，图片会拼接在一起，并撑满全屏。

这里提供一个案例，仅供参考。

第11章

SMIL电子书（听书）

本文介绍使用Media Overlays技术实现文本与音频的同步播放，也称之为听书。并提供一个完整SMIL电子书案例。

Media Overlays在ePub中指的是视觉内容与听觉内容同步呈现，即把预先录制好的音频内容与书面文字相匹配。

那Media Overlays是怎么工作的呢？其实很简单，给出一个文本片段并与之对应的音频片段做同步：

1. 第1个文本片段对应音频文件中0s到16s之间；
2. 第2个文本片段对应16s到33s；
3. ……

支持Media Overlays的阅读器，可以实现下面的三个功能：

1. 点击播放叠加的音频；
2. 正在播放的音频片段与之对应的文本突出显示；
3. 点击某个文本片段，切换到该文本的音频片段，然后播放。

这里的片段可以是一个整段、一句话甚至是一个字，你可以根据实际的用户体验采纳某一种方式。如果你手动生成的SMIL精度越高（段＞句＞字），你所需要做的工作就越多。

本章内容大部分是How To Create EPUB 3 Read Aloud eBooks这篇文章的译制内容，主要面向的需求是歌词与音乐同步播放。

11.1 SMIL文件语法

参考规范EPUB Media Overlays 3.1，下面代码展示使用SMIL文件描述文字与音频的同步信息：

```
<smil version="3.0" xmlns="http://www.w3.org/ns/SMIL" xmlns:epub="http://www.idpf.org/2007/ops">
  <body>
    <seq epub:textref="../Text/QingPinDiao.xhtml" epub:type="bodymatter chapter" id="QingPinDiao">
      <par id="fw01">
        <text src="../Text/QingPinDiao.xhtml#lrc01"/>
        <audio clipBegin="00:00:33.11" clipEnd="00:00:39.51" src="../Audio/QingPinDiao.mp3"/>
      </par>
      <par id="fw02">
        <text src="../Text/QingPinDiao.xhtml#lrc02"/>
        <audio clipBegin="00:00:39.51" clipEnd="00:00:45.83" src="../Audio/QingPinDiao.mp3"/>
      </par>
      <par id="fw03">
        <text src="../Text/QingPinDiao.xhtml#lrc03"/>
        <audio clipBegin="00:00:45.83" clipEnd="00:00:52.53" src="../Audio/QingPinDiao.mp3"/>
      </par>
    </seq>
  </body>
</smil>
```

这个SMIL文件的3个片段中的每一个都规定了4件事情：

1. 文本片段标识符（../Text/qpd.xhtml#lrc01）；
2. 文本所使用的音频文件（../Audio/QingPinDiao.mp3）；
3. 文本片段开始时间（00:00:33.11）；

4. 文本片段结束时间（00:00:39.51）。

上面的SMIL文件用中文来表述为：

在*QingPinDiao.xhtml*文件中的文本片段lrc01，即是音频文件中的33.11s到39.51s这一段内容。

其中每个SMIL文件仅引用一个XHTML文件和一个音频文件，具有简单的线性结构，如上例所示：

1. 一个`<seq>`元素，指定其子元素的播放序列；
2. 一组`<par>`元素，每个元素都指定`<text>`与`<audio>`节点之间的同步。

11.2　创建流程

创建的ePub文件必须是ePub 3.0标准，在创建SMIL文件之前，先排版好内容。

对于要附加音频的每个XHTML文件，您需要执行以下步骤：

1. 获取一个音频文件，本次试验为一首歌曲；
2. 将文本进行标识，每句一个id，与SMIL对应；
3. 创建一个SMIL文件，指定文本和音频之间的同步信息；
4. 将播放时突出显示文本的CSS添加到样式表中；
5. 将相关的元数据添加到电子书的OPF文件中。

步骤1：音频文件

本次案例使用的音频文件（心经）是从网页云音乐播放器上下载的，音频文件推荐使用MP3格式，采样44100 Hz或48000 Hz。

如果是一本书，你可以自行录制自己的声音，或使用免费无版权的音频文件。

步骤2：文本标记id

给XHTML文件中的文本片段添加id，本次演示的是歌曲，所以我采用一句歌词为一个粒度。

部分XHTML代码：

```
<h2><span id="xj01">般若波罗蜜多心经</span></h2>
<p><span id="xj02">观自在菩萨</span>，<span id="xj03">行深般若波罗蜜多时</span>，<span id="xj04">照见五蕴皆空</span>，<span id="xj05">度一切苦厄</span>。<span id="xj06">舍利子</span>，<span id="xj07">色不异空</span>，....，<span id="xj51">波罗揭谛</span>，<span id="xj52">波罗僧揭谛</span>，<span id="xj53">菩提萨婆诃</span>。」</p>
```

步骤3：确定歌词

然后使用foobar2000搜索歌词文件，这里推荐使用@Asion汉化的版

本，自带ESLyric歌词插件。

使用方法：

1. 安装foobar2000软件；
2. 打开从网易云音乐下载的歌词文件，播放音乐；
3. 依次点击「视图→ESLyric 面板」，在弹出窗口中，点击鼠标右键选择「搜索歌词」；
4. 从搜索出来的歌词列表中选择合适的版本；
5. 点击鼠标右键，依次选择「复制歌词→复制歌词」，粘贴到EmEditor中。

我这里使用的是这种版本，歌词粒度精确到每个字：

```
[ar:王菲]
[ti:心经(佛经唱诵)]
[al:Smile Angel]
[00:07.96]观[00:08.58]自[00:09.26]在[00:09.76]菩[00:10.69]萨
[00:11.14]
[00:14.82]行[00:15.64]深[00:16.87]般[00:17.39]若[00:17.89]波
[00:18.32]罗[00:18.94]蜜[00:20.30]多[00:21.12]时[00:21.62]
[00:25.00]照[00:25.73]见[00:26.92]五[00:27.35]蕴[00:27.75]皆
[00:28.73]空[00:29.30]
...
```

现在需要做的是，删除不需要的时间戳，参考我们使用的正文，进行断句。还需要添加章节名，即「般若波罗蜜多心经」一句。处理如下：

```
[00:00.00]般若波罗蜜多心经
[00:07.96]观自在菩萨
[00:14.82]行深般若菠萝蜜多时
[00:25.00]照见五蕴皆空
...
```

然后使用正则表达式给时间戳与文字间加空格，正则表达式：

```
查找：\]([\u4e00-\u9fa5])
替换：]\t\1
```

这一步的作用是粘贴到Excle中可以分列显示。替换完成后如下：

```
[00:00.00] 般若波罗蜜多心经
[00:07.96] 观自在菩萨
[00:14.82] 行深般若菠萝蜜多时
[00:25.00] 照见五蕴皆空
...
```

步骤4：生成播放序列

然后开始参考上一节的SMIL文件语法，添加播放序列和文本片段等信息。因为id不可以使用纯数字命名，所以我们需要再id前添加字母内容。复制替换好的歌词文件到Excle中，处理如下：

播放序列	文本片段	开始时间	文本内容	结束时间
play01	xj01	[00:00.00]	般若波罗蜜多心经	[00:07.96]
play02	xj02	[00:07.96]	观自在菩萨	[00:14.82]
play03	xj03	[00:14.82]	行深般若菠萝蜜多时	[00:25.00]
play04	xj04	[00:25.00]	照见五蕴皆空	[00:32.67]
play05	xj05	[00:32.67]	度一切苦厄	[00:44.67]
play06	xj06	[00:44.67]	舍利子	[00:48.37]
play07	xj07	[00:48.37]	色不异空	[00:50.68]
play08	xj08	[00:50.68]	空不异色	[00:59.66]
play09	xj09	[00:59.66]	色即是空	[01:01.47]
play10	xj10	[01:01.47]	空即是色	[01:10.39]
play11	xj11	[01:10.39]	受想行识	[01:14.06]
play12	xj12	[01:14.06]	亦复如是	[01:27.49]
play13	xj13	[01:27.49]	舍利子	[01:29.68]

图11-1　smil语法内容

一共是53个文本片段，这里截取部分显示。最后一句就是歌曲的截止时间，别忘记添加。第一句的结束时间是第二句的开始时间，以此类推，最后一句即是音频的结束时间。这样才可以无缝衔接起来，连续播放。

使用正则表达式统一格式化为SMIL信息：

```
查找：(.*?)\t(.*?)\t\[(.*?)\]\t([\u4e00-\u9fa5]{1,})\t\[(.*?)\]\n
替换：<par id="\1"><text src="../Text/Chapter2-2.xhtml#\2"/><audio clipBegin="\3" clipEnd="\5" src="../Audio/XinJing.mp3"/></par>\n
```

替换好的<par>标签如下：

```
<par id="play01">
  <text src="../Text/Chapter2-2.xhtml#xj01"/>
  <audio clipBegin="00:00.00" clipEnd="00:07.96" src="../Audio/XinJing.mp3"/>
</par>
<par id="play02">
  <text src="../Text/Chapter2-2.xhtml#xj02"/>
  <audio clipBegin="00:07.96" clipEnd="00:14.82" src="../Audio/XinJing.mp3"/>
</par>
<par id="play03">
  <text src="../Text/Chapter2-2.xhtml#xj03"/>
  <audio clipBegin="00:14.82" clipEnd="00:25.00" src="../Audio/XinJing.mp3"/>
</par>
...
```

然后参考上一节的SMIL文件语法，补齐首尾代码，完成后如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<smil version="3.0" xmlns="http://www.w3.org/ns/SMIL" xmlns:epub="http://www.idpf.org/2007/ops">
  <body>
    <seq epub:textref="../Text/Chapter2-2.xhtml" epub:type="bodymatter chapter" id="XinJing">
      <par id="play01">
        <text src="../Text/Chapter2-2.xhtml#xj01"/>
        <audio clipBegin="00:00.00" clipEnd="00:07.96" src="../Audio/XinJing.mp3"/>
      </par>
      ...
      <par id="play53">
        <text src="../Text/Chapter2-2.xhtml#xj53"/>
        <audio clipBegin="05:44.62" clipEnd="06:16.29" src="../Audio/XinJing.mp3"/>
      </par>
    </seq>
  </body>
</smil>
```

命名与文本XHTML一样，然后添加SMIL格式类型后缀。在Sigil中将这个文件导入到Misc文件夹中，然后导入歌词文件到ePub中。

```
Chapter2-2.xhtml.smil
```

步骤5：修改CSS

为同步播放时突出显示的文本内容添加样式：

```
.-epub-media-overlay-active {
    background-color: #FFFF99;
}
```

步骤6：修改OPF文件

除了常规的元数据选项，比如书名、作者等，还需在OPF文件中添加一些其他元数据。

在OPF文件的<metadata>部分，你需要给电子书的音频文件指定一个音频持续时间。然后给所有音频设置一个总时长。

*Chapter2-2.xhtml*章节的音频时长：

```
<meta refines="#XinJing" property="media:duration">06:16.29</meta>
```

总时长：

```
<meta property="media:duration">06:16.29</meta>
```

给音频文件设置一个演唱者：

```
<meta property="media:narrator">Faye Wong</meta>
```

定义上一个步骤使用的类选择器：

```
<meta property="media:active-class">-epub-media-overlay-active</meta>
```

然后是`<manifest>`资源清单项目。定义*Chapter2-2.xhtml*文件叠加的媒体文件：

```
<item id="Chapter2-2.xhtml" href="Text/Chapter2-2.xhtml" media-type="application/xhtml+xml" media-overlay="XinJing"/>
```

音频文件：

```
<item id="XinJing.mp3" href="Audio/XinJing.mp3" media-type="audio/mpeg"/>
```

SMIL文件：

```
<item id="XinJing" href="Misc/qpd.xhtml.smil" media-type="application/smil+xml"/>
```

> 每个XHTML文档最多可以与一个SMIL文件相关联。
> SMIL文件的`<item>`条目id一定要与media-overlay属性的内容一致。

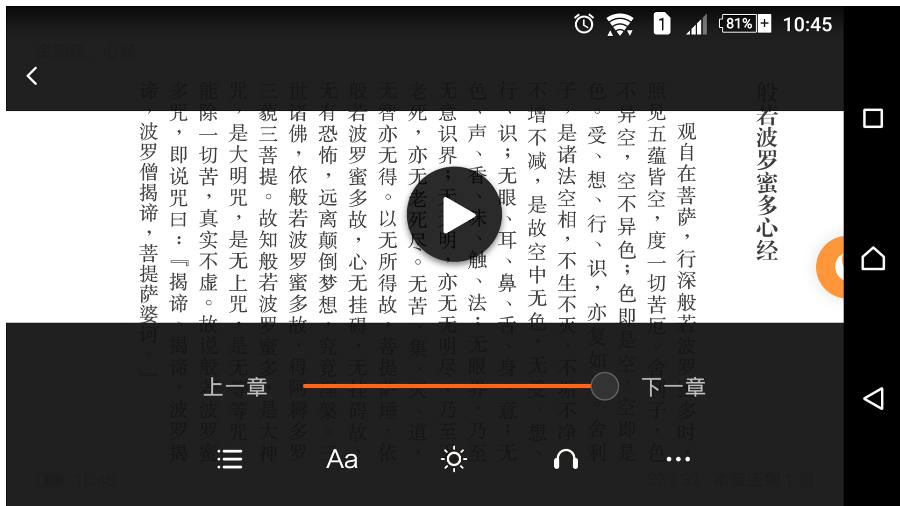


图11-2　点击屏幕中央，显示播放按钮



图11-3　同步播放的文本突出显示

《心经》案例下载，这个文件里还有一个未做的《金刚经》。

11.3　使用Audacity标记时间

上面演示的都是歌曲，本身已经有时间戳了。如果是自己录制的音频则需要使用专业工具处理。

为了使每个文字与其对应的音频片段匹配，对于已经录制好的音频文件需要手动逐字进行标记。也可以根据需要以词或句为单位进行标记。为了能够准确定位出每个文字的音频片段，推荐使用Audacity音频处理软件对音频进行标记。

其中，文字的标记工作需要人工手动完成，可利用该软件通过查看音轨波形来切分，一般来说，一个字或词的波形会出现明显的尖峰，将其选中，利用该软件对选中的音频添加时间标签（快捷键为Ctrl＋B）。待全部切分完毕后，软件可以自动导出这些标签并生成TXT文件（文件→导出标记）。

下面以 @白云出岫 老师朗读的李煜《虞美人·春花秋月何时了》为例。

打开音频文件

依次点击「文件→打开」，找到下载的音频文件，然后点击「打开」按钮。打开后的音频文件，会显示波形，可以按住Ctrl键，然后滚动鼠标滚轮，放大或缩小波形。

添加时间标记

在添加时间标记之前，先从网络上获取朗读的文本，然后听一遍朗读，注意观察波形的变化。本次演示一句一个标记，所以我们特别需要注意断句的地方。

在听第2遍时，在断句的位置按快捷键Ctrl＋B添加标记。该示例断句部分，完全是一条直线。

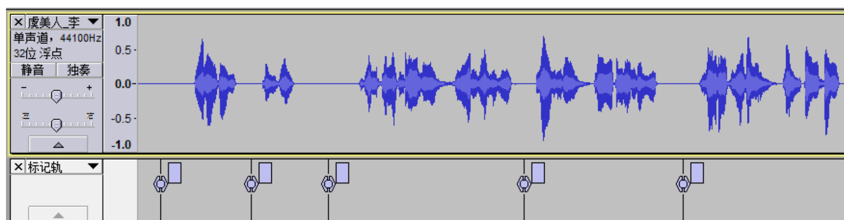


图11-4　添加停顿处标记

每句的「开始时间」标记完成后，在标记框中输入文本：

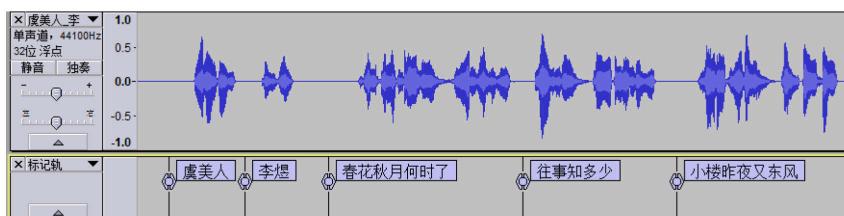


图11-5　停顿处标记命名

如果是一本小说，添加标记的方式可以是数字编号，而不是演示中的正文文字。

处理时间标记

依次点击「文件→导出标记」，在弹出对话框中选择保存文件夹，点击「保存」按钮，默认使用「标记轨」命名的TXT文本文档。使用EmEditor打开刚才保存的「标记轨.txt」文本。打开文本之后，先不要处理，回到Audacity软件中，查看音频结束时间，同时记录这个时间。这是因为我们上一步只标记了每句的开始时间。

1	0.836870	0.836870	虞美人
2	2.902894	2.902894	李煜
3	5.178136	5.178136	春花秋月何时了
4	10.460880	10.460880	往事知多少
5	14.645233	14.645233	小楼昨夜又东风
6	19.457238	19.457238	故国不堪回首月明中
7	24.923048	24.923048	雕栏玉砌应犹在
8	29.395074	29.395074	只是朱颜改
9	33.527122	33.527122	问君能有几多愁
10	38.234518	38.234518	恰似一江春水向东流

图11-6　每句起始时间

「0.836870」时间标记表示文本「虞美人」的开始位置，其中0表示秒，836表示毫秒。再观察上图，可知文本「虞美人」的开始时间为0.836870，结束时间为2.902894，但实际上我们应该从0.000000开始计算第一句文本，所以文本「虞美人」的正确开始时间是0.000000；后面的文本「李煜」开始时间为2.902894，结束时间5.178136，然后以此类推。最后一句「恰似一江春水向东流」的结束时间即是音频文件的结束时间45.923000。

参考上一节的内容，生成SMIL文件，这里不再叙述。

第12章

不同格式的文档制作ePub

电子书格式多种多样，普通用户可以接触到的，也就几种，见1.4节。而这些格式的电子书在做成ePub格式的过程中，需要用到不同的方法和工具，然后才可以制作成ePub格式。中间的媒介，一般是TXT格式。这种格式是最基础也是最干净的，只包含文字信息，我们处理起来最方便。然后是ePub格式，我们可以在原有的基础上再修改排版，使样式变得精美。TXT和ePub就是我们在处理不同格式的电子书时，要获得的文档格式。

本章的主要内容为不同格式类型的电子书如何处理。一般需要处理以下类型：图像PDF、双层或矢量PDF、Word文档、TXT文本、EPUB电子书、Kindle电子书（MOBI/AZW3格式），这6种类型的资源，基本上就是我们制作电子书中遇到的绝大多数格式了。其中矢量PDF、Word、TXT、MOBI、AZW3可以快速的转成ePub格式。所以，制作ePub电子书又分为OCR和修改两部分，OCR针对的是图像PDF，修改指的是ePub排版。

下面这张图介绍了，针对不同类型的电子书，使用什么方法和工具，可以把它们转成TXT和ePub格式。

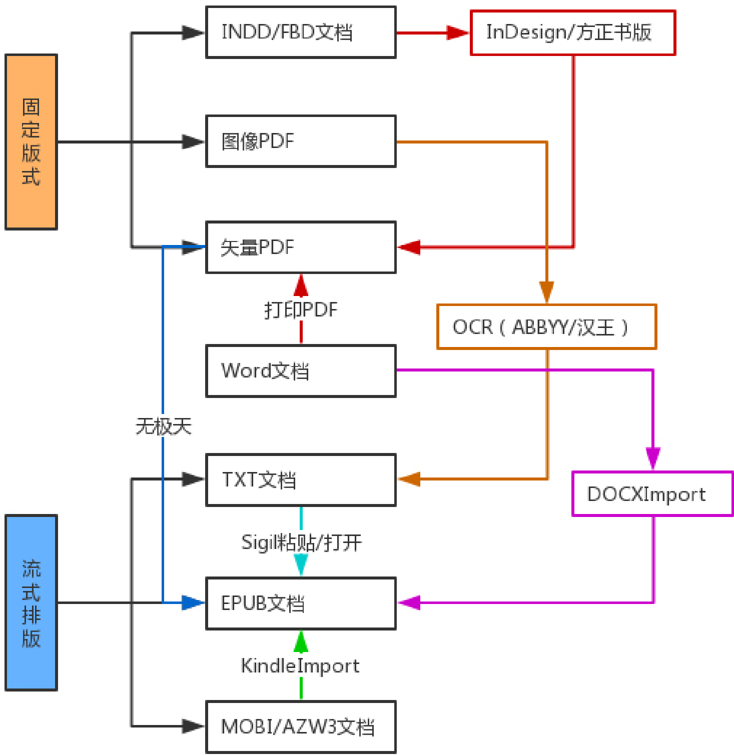


图12-1　不同类型的文档如何处理

12.1　排版文件与版式文件

在介绍不同类型的文档处理之前，先普及下一些基础知识。

在传统出版流程中，产生了各种格式的电子排版文件。例如，图书排版用方正书版软件，生成*.fbd小样；报纸（也能用来排版图书）用方正飞腾，生成*.fit文件；期刊（同样也可以排版图书）用Adobe Indesign，生成*.indd文件；办公文件用MS　Word，生成*.docx文件等。这些文件可以统称为排版文件，是用排版语言写成的排版命令集合。

排版文件在用于打印/印刷时，需要转换为版式文件，格式有PDF、PS（PostScript）、大样（如S92等）。版式文件是排版命令执行的最终结果。打印/印刷设备只能解析版式文件而不能解析排版文件，这是因为版式文件的格式是各种不同打印/印刷设备通用的，而排版语言则是各个排版软件自己特有的。排版文件是可编辑、可重排的，而版式文件一般是不可再改变的。PS、PDF都是世界通用、公开的标准，大样则是方正曾经的过渡格式。从排版文件到版式文件，一般只能由相应的排版软件提供输出功能，因为排版语言是该排版软件特有的、不公开的。例如，方正书版2008或更新版本可将FBD、S92输出PDF，Adobe Indesign可将INDD输出PDF，MS　Word可将DOC输出PDF，方正飞腾可将FIT输出PS，Adobe　Distiller可将PS转PDF。对于没有电子排版文件的纸质历史数据，则可以通过扫描成图像——OCR——校对，生成双层PDF版式文件或TXT格式。

显然，排版文件由于其专有性，并不能直接用于数字出版。版式文件中，大样文件由于是方正私有的、过渡的，同样不适合。PS文件虽然是公开标准，但是由于过于复杂，缺乏软件支持，也不合适。PDF文件则有很多很好的特性，例如Adobe及其他厂商提供的丰富处理软件和免费阅读器，良好的压缩及阅读性能等，是数字出版的重要格式。但是，PDF并不能满足数字出版的全部甚至主要需求。数字出版的需求，包括电子书阅读和数据库应用。电子书阅读包括版式阅读和流式阅读，数据库应用包括资源数据库和知识数据库。PDF可以满足版式阅读的需求，

但是不能满足流式阅读、数据库应用的需求。即使是版式阅读，PDF也不是最佳的方案，很多问题只有版式HTML才能解决。

以上内容见无极天网站。

12.2 TXT文本制作ePub

本次演示以适配多看阅读为例，其他阅读器可根据自身特性，进行细节的修改。

在使用TXT格式的文本制作ePub之前，我们先在网络上搜索下这本书是否有PDF文件，最好是原书扫描版的。在处理TXT文本时，PDF可以作为参考校对。而且PDF文档因为是纸书的扫描版，所以可以保留排版样式信息。这个在制作ePub时可做为参考。

如果你实在找不到PDF文件，可以使用全国图书馆参考咨询联盟这个网站，预览试读页。使用方法见8.1.1节，虽然不能浏览全文，但是因为每章的样式都是统一的，所以只看前面20页就能知道排版怎么做了。

制作过程中需要创建一些文件夹用来存放制作过程中产生的文件，推荐如下：

```
书名
├─书籍信息
├─插图
└─文本与PDF
```

「书名」文件夹下有三个文件夹：书籍信息、插图、文本与PDF。分别叙述如下：

1. 书籍信息：存放有关本书的资料，比如封面图片、版权页等；
2. 插图：文前彩页、文中插图等；
3. 文本与PDF：源文本、扫描版PDF或辅助页试读页PDF等。

加工生成的epub文件就保存在「书名」文件夹下。

常用网站

亚马逊：查询目录信息、封面、版权页；

全国图书馆参考咨询联盟：查询内容简介、关键字、中图法、封面、辅文页、正文试读页（前20页）；

中国版本图书馆：查询内容简介等。找到该书的版权页，输入CIP核字号，然后查询。

新浪微盘：这个网盘可以使用搜索语法，在微盘分享的文件中搜索某一文件。例如：中国的类书、政书和丛书 PDF site:vdisk.weibo.com

搜索图书信息，除了电商网站之外，还可以到出版社官网找找，如果出版社有网站的话。

12.2.1　文档整理

每本书虽然内容不同，但是制作流程基本相同。比如TXT文档，在制作前，我会把文件编码转成Unicode编码、段首无缩进、段落间空一行等。复杂一些的就是在文本整理时就处理好相关样式，给不同段落内容添加符号进行区分，然后使用搜索模板批量处理。这些制作前的准备工作就是这节要说的内容。

工艺设计是指根据适配的阅读器而确定的版式要求，工艺设计可以参考纸书的版式自己确定。后面排书就按照这个工艺执行，保证风格统一，电子书能适配多个阅读器。这需要由排版经验丰富的操作人员，在多个阅读器中反复试验，总结一套通用的标准。

整理原稿

整理文本的目的是为了更快地制作ePub电子书，它只是一个过渡！

在制作一本电子书中，因为是从一种格式或载体转换为另一种格式，在这个转换或者传播的过程中，时常会因为编码或格式转换的原因造成一系列错误，比如敏感字、字符乱码、断行等。所以我们在制作电子书之前就要先对文本进行规范化整理。有如下要求：

1. 标题层级安排有序，格式统一。比如「第一章标题内容」中，「章」字后面应该有一个空格做区分的。这里的「章」也可以是节、部、卷等。
2. 目录不能有遗漏，可以在生成目录后对照原文档（如纸书或扫描的PDF文件等）检查。
3. 文本无缺失，无误删。无法确定是否有意义的内容，建议保留，不要删除。网文敏感词，可根据上下文语境，阅读补充。
4. 段落应该是正常结束的。对于段落在内容还没讲完就分段的，使用正则表达式查找出来后，参考原文档合并段落。
5. 对于数字和字母，使用半角字符显示。
6. 中文书标点应是全角形式。
7. TXT文本无首行缩进，无空行（如果是用Sigil打开TXT文本，请保留空行）。

如果你是读校一本书（在PC端，格式为Word或TXT），可以在读校中参照PDF标记好要制作的样式，然后再导入Sigil中，用正则表达式查找替换样式。文本整理的示例见7.2节内容。至于标记的方法，可以自己确

定，也可以参考下面的markdown语法。

電電markdown

电电转换器官网，这是一个台湾网站，如果你不能访问，请自行寻找解决办法。

该网站主要通过对文本文档初步加工，然后使用转换工具，输出高质量的ePub电子书。适用于以文字为中心的内容，不适合用于以图片为中心，要求具有复杂版面的内容制作。

在本书的5.4.2一节中，可以得知Sigil是可以直接打开TXT格式文档的，但需要处理如下：

1. 文件编码是UTF-8或Unicode，ANSI无效；
2. 段落无缩进，即首行不要有空格；
3. 段与段之间需要有空行。

如果你想使用Sigil打开TXT文档时，以上3点是必须要处理的。下面就这3个问题，分别给出解决办法。

修改文本编码

使用系统自带的记事本软件打开要制作的文本文档，选中文件，点击鼠标右键选择「打开」或直接双击该文本文档。文本文档一般默认是用记事本软件打开，如果你修改了默认打开软件，请在打开方式里选择记事本软件打开该文件。

这里使用的是中国文化史知识丛书中的第2本《中国的类书、政书和丛书》，全书文本可以在中国知网的数字出版物超市中免费获得。《中国的类书、政书和丛书》PDF版本。

文本打开之后，依次点击「文件→另存为」：



图12-2　另存为TXT文件

点击「另存为」选项后会弹出一个对话框，可以在这个对话框设置要保存的文件名、编码、保存位置等。为了与源文本区别开来，另存为的文本我们在文件名前添加编码名称，这里以Unicode为例。然后在编码中选择以Unicode编码储存文本：

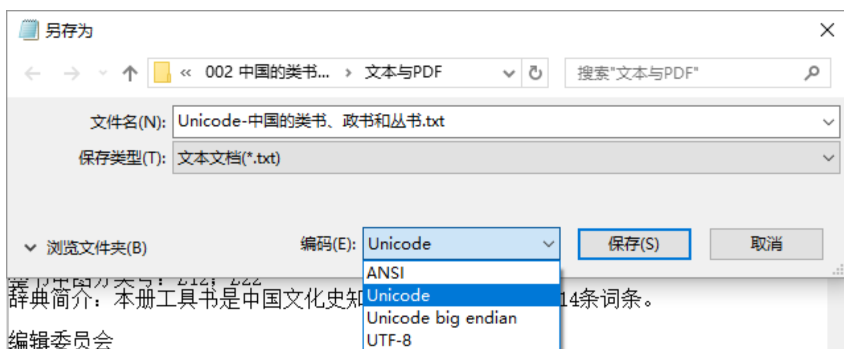


图12-3　选择TXT文本的编码

确认没有问题后，点击「保存」按钮。在源文本所在的文件夹中会出现一个新的文本文档，新的文本文档与源文本的区别是，新文本前面多了Unicode编码名称。

清除首行缩进

在7.5一节中，我已经介绍过如何使用宏脚本。删除首行缩进我们就使用精校组排版脚本.jsee这个脚本文件中的功能。

使用EmEditor软件打开刚才另存为的TXT文本文档，然后安装「精校组排版脚本.jsee」，教程见「7.5」一节。安装完成后，运行这个脚本：

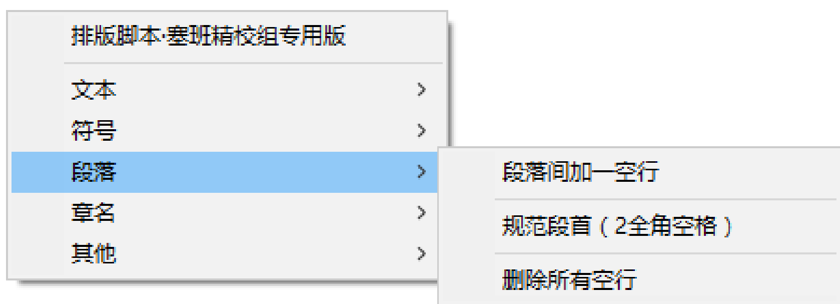


图12-4　处理段落格式

依次点击「段落→规范段首（2全角空格）」，点击之后，宏脚本会自动开始处理当前打开的文本文档。完成后以后，每个段落的首行都会添加2个全角空格。

你可能发现我们的需求不是删除首行缩进吗？为什么要先添加首行缩进？其实，从网络上获取的文本文档一般是没有经过软件格式化过的，可能有的段落缩进了，有的没有，有的缩进了2个全角空格，有的缩进1个，也有可能缩进的是英文半角空格。执行规范段首的意义是把所有可能的情况都统一为一种形式，即首行缩进2个全角空格。这样便于我们后续的处理。

然后使用正则表达式批量删除首行的2个全角空格，正则表达式语法如下：

```
查找：^　　
替换为空
```

^字符后面紧跟2个全角空格。

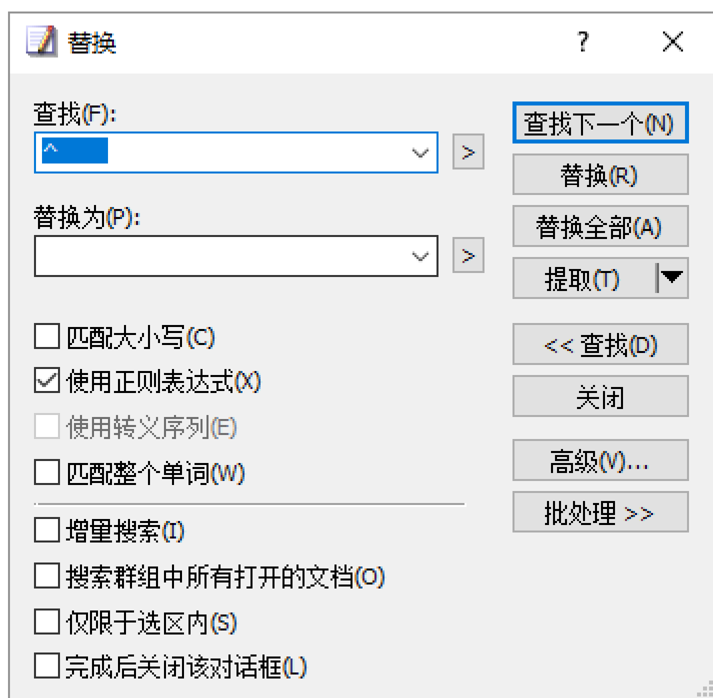


图12-5　删除段首空格

点击「替换全部」按钮，删除当前文件中的首行2个全角空格。

段落间添加空行

运行「精校组排版脚本.jsee」宏脚本，依次点击「段落→删除所有空行」。同样，先清除不规范的段落空行，把所有段落统一成一种模式。然后依次点击「段落→段落间加一空行」，完成后每个段落见都会添加

一个空行。如下：

↓
类书有多种类型：↓
↓
从编录的体裁看，最早的形式是征事的，如《皇览》；有征事兼采诗文的，如《艺文类聚》；有辞藻典故的，如《佩文韵府》；有编成韵语的，如《幼学故事琼林》；有重视图表、图文并茂的，如《三才图会》、《图书编》；还有几种综合的等等。↓
↓
从类书编纂的情况看：有官修的，是用国家的力量集中人力物力编纂大规模的类书；有文人学者为作诗写文储备材料而自己编印类书的；待印刷发达、刻板盛行、书肆林立后，书坊也编辑类书。↓
↓
从类书的编录方法来看，分类是主要形式，而且大多数类书都是采取分类编录的形式。后来又发明了采用“字典”式的依韵编录的方法。这种类书，已消失了分类的形式，但仍有百科全书和资料汇编的实质；所以，没有分类形式的类书仍属类书，如最著名的《永乐大典》即是。还有以数目字来编录的，专以数字为纲来聚集事实，如《小学绀珠》；这种类书为数极少，也就不做介绍了。↓
↓
为论述方便起见，我们从类书的内容性质，依其采辑范围，把类书概括为两大类：↓
↓
（1）汇编各种材料，山包海涵，综合各类的一般类书——这是类书的正宗，也就是综合性类书。↓
↓
（2）只辑一类的专门类书——这是类书的别体，也就是专门性类书。↓
↓

图12-6　添加段落空行

3点要求完成以后，我们就可以开始整理文本内容了，主要是标点、空格、断行等问题。

12.2.2　内容整理

内容整理指的是用宏脚本先把英文半角符号批量转成中文标点、使用正则表达式查找非标准换行、不必要的空格等。

对于中文段落，除去数字中间的逗号分割符，使用英文逗号后加半角空格隔开以外，标点符号必须使用中文标点符号，即全角形式。波浪线使用中文全角波浪线，破折号完整。

对于英文段落的标点符号全部使用英文标点符号，分句符号（如：逗号，句号，感叹号等）需要特别注意后面加一个半角空格符号。两个相邻标点之间不需要加空格。

半角标点转全角

运行「精校组排版脚本.jsee」宏脚本，依次点击「符号→半角标点→全角」。或者使用「规范标点符号.jsee」宏脚本处理文本。

查找断行

见4.4节，使用正则表达式查找文本的非标准换行。不能确定是不是断行

的，可以参考PDF文件。

空格

参考原书PDF文件，处理TXT文档。比如标题序号与标题内容之间的空格：

一 类书
1 什么是类书？
2 类书的出现和发展
3 类书的作用
4 历代都有哪些重要类书？
5 结束语

建议使用半角空格形式，这种整理好的格式，都可以使用正则表达式查找出来，可以统一格式化排版。

比如这个丛书特有的拼音标注：

棣〔dì 第〕 胤禛〔yìn zhēn 印真〕 蹴〔cù 促〕……

完成上述步骤之后就可以使用Sigil打开这个文本，开始加工成ePub电子书。

12.2.3 拆分正文

以上所有步骤完成以后，我们就可以正式的开始使用Sigil来加工这个文本文档了。然后是确定好ePub版本，这里使用2.0标准。

导入TXT文本

打开Sigil主程序，依次点击「文件→打开」，在弹出窗口中选择文本编码为Unicode的文档：

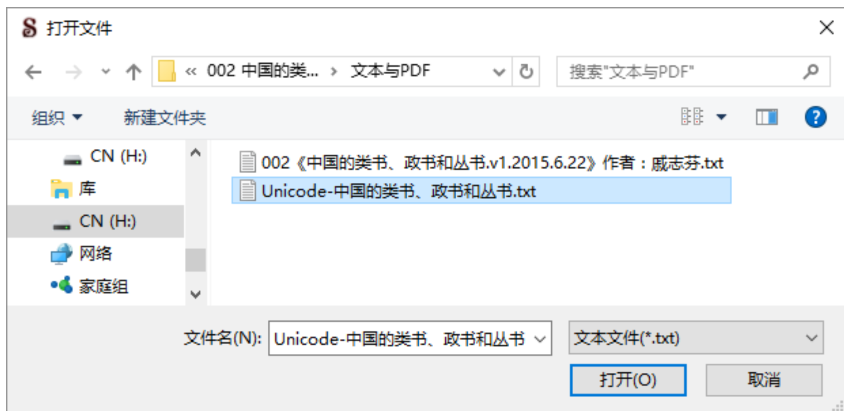


图12-7　使用Sigil打开TXT文件

注意右下角的文件类型为「文本文件」，然后点击「打开」按钮。TXT文本文件打开后，显示如下：

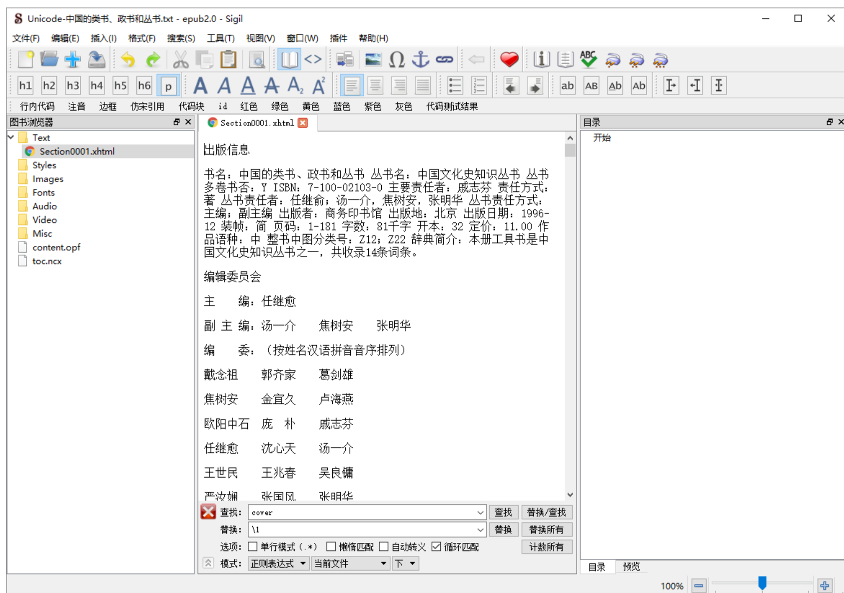


图12-8　已打开的TXT文件

然后可以执行一次「保存」，依次点击「文件→保存」或使用快捷键Ctrl＋S。文件命名如下：

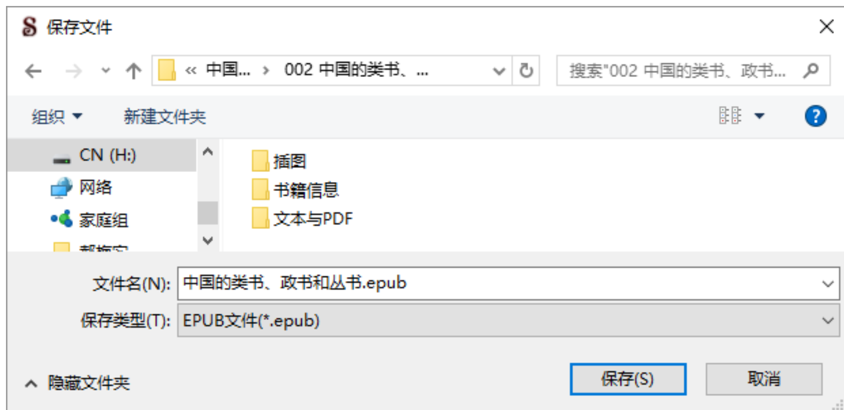


图12-9　保存为epub文件

点击「保存」按钮，保存epub文件。

格式化标题

我们先处理正文内容，辅文内容在目录生成之后再处理。按快捷键F2或依次点击「视图→代码视图」，我们使用正则表达式批量添加拆分标记。

在添加拆分标记之前，我们先确定下规则，即我们想要做到什么效果？参考PDF文件，第1个一级标题前的内容单独一个HTML文件、一级标题单独一个HTML文件、二级标题及内容单独一个HTML文件。根据这个规则我们就可以先对一级标题和二级标题进行标题格式化，然后在二级标题前添加拆分标记。

标题格式化见5.4.8节，比如编辑委员会、编者献辞、内容简介和作者简介、引言等就可以使用工具栏中一级标题按钮格式化，剩下的正文一级标题使用正则表达式批量处理，这是因为一级标题、二级标题都是有规律的文本，所以可以用正则表达式查找出来，一次性修改。同时这些标题刷了h1的标题标签之后，点击工具栏中的拆分按钮或者依次点击「编辑→在光标处拆分」。完成后如下：

Text
Section0001.xhtml
Section0002.xhtml
Section0003.xhtml
Section0004.xhtml
Section0005.xhtml
Styles
Images
Fonts
Audio
Video
Misc
content.opf
toc.ncx

引言

我国是个历史悠久的文明古国，自有文字以来，积累了丰富的文化典籍。尤其是我们的祖先发明了纸、雕版印刷以及活字印刷之后，中国的文化事业更加突飞猛进地发展，各类图书，浩如烟海。这些典籍集中反映了中国文化的灿烂辉煌。它们是中华民族智慧的凝聚，知识的结晶，成了我国的宝贵文化遗产。

我们知道，书籍从使用的角度看，可分为两大类，一是为了求知识或欣赏，需要从头到尾细细阅读的；二是当我们遇到疑难需要解决时，才去翻检或引用的工具书。工具书是以特定的编排形式和检索方法编制的书。这种书是人们在学习或做科学研究中所经常使用的，就像人们进行生产或其他活动时离不开必要的工具一样。此外，在我国古籍中还有不少大部头的书，其本身不是工具书，但因部头太大，一般并不要求人们从头到尾把它们读完，往往是利用它们翻检材料的，所以，它们也就有了工具书的性质。这种书介于阅读和翻检之间，即通常所谓的“资料书”；现在人们把这些书称为“广义的工具书”。在国外，有人称之为“边缘工具书”。

工具书、广义工具书的种类都很多，决不是这本小册子能够介绍得完的。现仅选择其中的类书、政书和丛书予以阐述，因为它们毕竟都是我国花团锦簇的文化典籍中纷呈异彩的花朵，同样都在我国文化史上闪耀着各自的霞光。

一 类书

图12-10　拆分内容文档

拆分章节

切换到代码视图中，依次点击「工具→HTML重新格式化→Mend and Prettify All HTML File」，重新格式化HTML标签代码，这一步是为了消除在代码视图中<p>后面的空格。在代码视图中，在需要插入拆分标记的位置插入光标，依次点击「插入→拆分标记」，完成后会出现一个拆分标记的代码：

```
<hr class="sigil_split_marker" /><p>一 类书</p>
```

获得了拆分标记代码之后，使用正则表达式同时格式化标题和添加拆分标记，语法如下：

```
查找：<p>([\x{4e00}-\x{9fa5}]{1}) (.*?)</p>
替换：<hr class="sigil_split_marker" /><h1>\1 \2</h1>
```

正则表达式语法解释：

- ([\x{4e00}-\x{9fa5}]{1})：表示一级标题中的一、二、三，即标题序号；
- (.*?)：表示的是一级标题中的标题内容；
- \1：即查找中的([\x{4e00}-\x{9fa5}]{1})；
- \2：即查找中的(.*?)。

正则表达式写完之后，查找一个匹配对象，点击「替换」按钮，查看替换有没有出错，无误后，点击「替换所有」，进行全文替换。

然后是二级标题的格式化和拆分标题添加，正则表达式语法：

```
查找：<p>(\d{1,2}) (.*?)</p>
替换：<hr class="sigil_split_marker" /><h2>\1 \2</h2>
```

正则表达式语法解释：

- (\d{1,2})：表示二级标题中的1、2、3、4、……，{1,2}表示最少是1位数，最多是2位数；
- \1：即查找中的(\d{1,2})；
- \2：即查找中的(.*?)。

拆分标题添加完之后，先别急着执行拆分动作。检查一级标题与二级标题之间是否也有内容，比如案例的第三章下就有3段介绍性的内容。如果遇到这种情况，一般我们会把这3段内容单独放在一个HTML文件中，所以就需要再一级标题下再插入一个拆分标记。

拆分标记添加之后，依次点击「编辑→在章节标记处拆分」或使用快捷键F6。拆分完成后，先保存。然后拆分完成后界面如下：

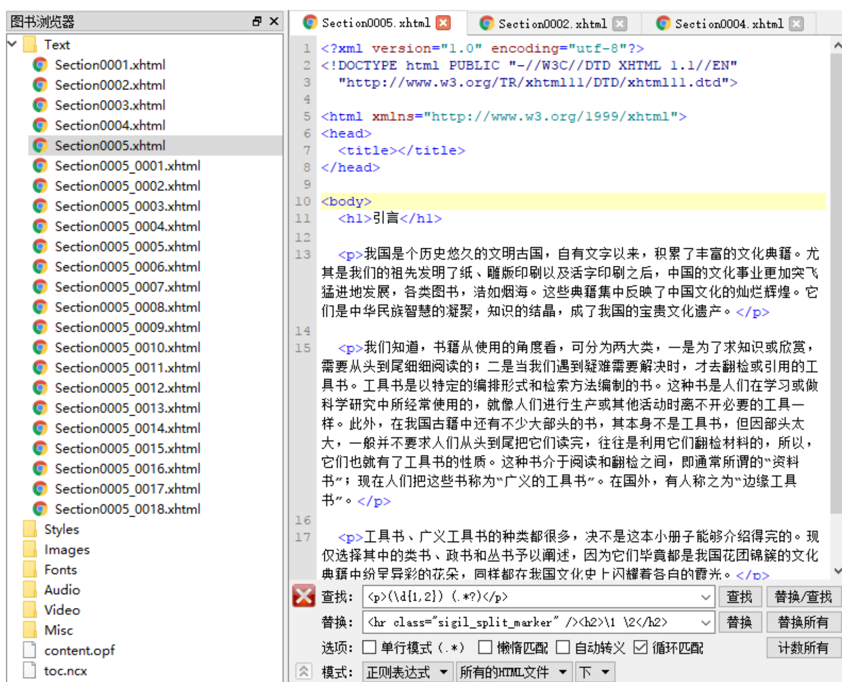


图12-11　批量拆分内容文档

拆分完成后，还需要根据内容来命名HTML文件，见下一节。

12.2.4　生成目录

上一节中讲了如何拆分一个HTML文件为多个HTML文件，然后拆分后

的文件还需要根据内容来进行命名。本书涉及的文件名如下：

- 封面页　Cover
- 版权页　Copyright-page
- 内容简介和作者简介　Introduction
- 编辑委员会　Contributors
- 文前彩页　Color-page
- 编者献辞　Foreword
- 第1个一级标题前内容　Preface
- 一级标题　Chapter
- 一级标题后二级标题前内容　Topic-sentence
- 二级标题　Subchapter
- 后记　Epilogue
- 附录　Appendix
- 版本记录　Colophon

参考上面的要求给HTML文件命名，同时删除不需要的文本，如出版信息（制作版权信息模板）、目录文本，最后的辅文图片提到「编者献辞」之前。更多命名规范见附录一一章。

然后使用Sigil的「添加封面」功能添加封面，见5.4.6节。封面图片可以从网络中找，然后处理标准如下：

1. 命名：cover
2. 格式：jpg
3. 尺寸：720×960或1440×1920
4. 单位：像素
5. 分辨率：72或96

封面的处理见6.3节内容。

学习5.4.9节，生成目录。确认要保留的标题层级，如下：

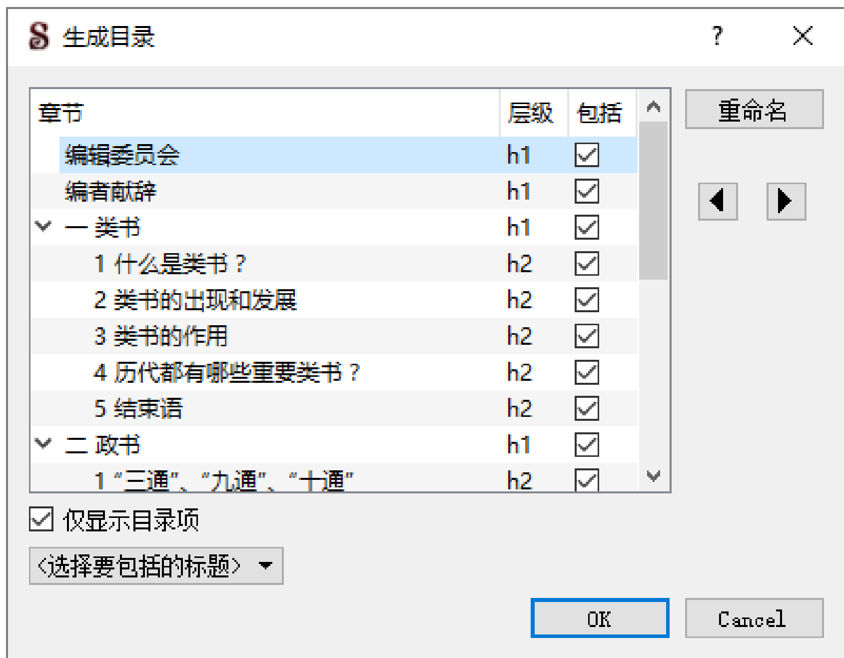


图12-12　生成目录

这里保留一级和二级标题，生成目录的依据是标题标签，有些网页没有标题标签，所以没有在目录中出现，我们需要手动添加。见5.3.10节编辑目录。

在制作一本电子书时建议添加一个记录版本信息的HTML文件，我们并不能一次到位就做出一本完美没有问题的电子书，所以有一个版本记录就非常有必要了。

目录编辑后如下：

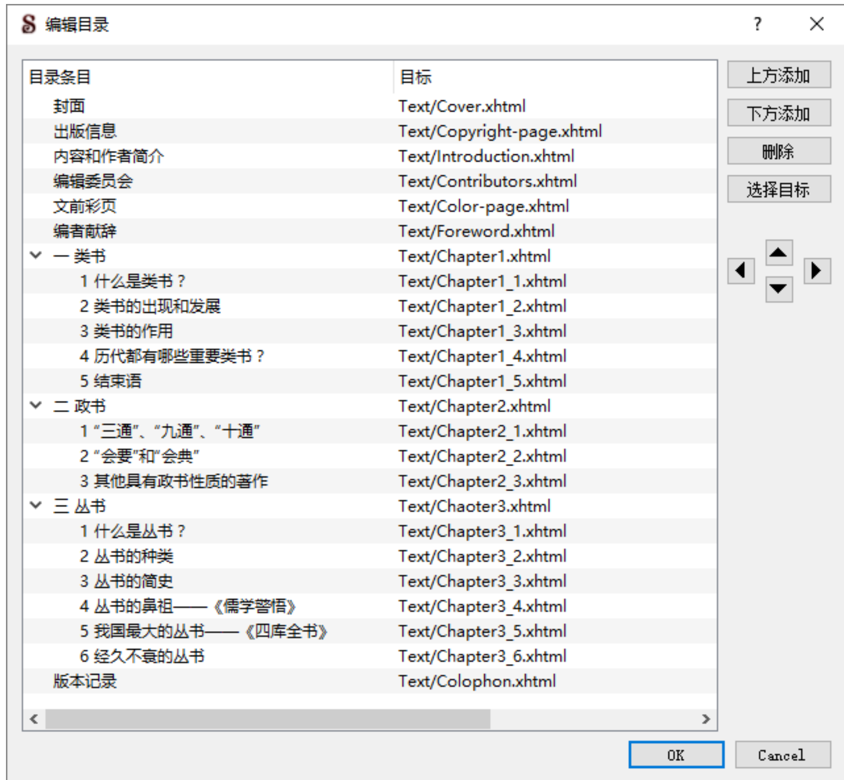


图12-13　编辑目录

现在目录已经完成了，后面我们就不需要编辑目录了，除非有大的变动。

12.2.5　封面和版权

本次演示旨在制作适配多看阅读的ePub电子书，见10.5.1给封面添加扩展代码。其中HTML文件中的代码，不用修改，仍使用Sigil的预置代码。然后在OPF文件中添加扩展属性。

然后是版权页，该示例属于一个系列，所以建议制作一个版权信息模板。HTML代码如下：

```
<body class="copyright-bg">
  <h1 class="copyright-title">出版信息</h1>
  <p class="BookInfo">书名：中国的类书、政书和丛书·002</p>
  <p class="BookInfo">丛书名：中国文化史知识丛书</p>
  <p class="BookInfo">ISBN：7-100-02103-0</p>
  <p class="BookInfo">作者：戚志芬</p>
  <p class="BookInfo">主编：任继俞</p>
  <p class="BookInfo">出版社：商务印书馆</p>
```

```
    <p class="BookInfo">出版日期：1996-12</p>
    <p class="BookInfo">字数：81千字</p>
    <p class="BookInfo">中图分类法：Z12；Z22</p>
    <p class="copyright">本书内容来源于互联网，未经作者授权，仅供读者学习及交流电子书排版技术。请于下载后24小时内删除！</p>
</body>
```

在「Styles」文件夹新建一个样式表，命名为「Stylesheet.css」，具体步骤见5.3.21一节。然后给除封面外链接这个样式表。开始书写具体的样式：

```
/*版权页背景*/
body.copyright-bg {
    background-image: url("../Images/chapter.png");
    background-size: 100% auto;
    background-repeat: no-repeat;
    background-position: bottom center;
}
/*版权标题*/
h1.copyright-title {
    margin: 0 0 0.8em;
    font-family: "小标宋";
    font-size: 1.35em;
    color: #ad3541;
    text-indent: 0;
}
/*图书信息*/
p.BookInfo {
    margin: 0;
    font-family: "黑体";
    font-size: .875em;
    line-height: 1.5em;
    color: #28363E;
    text-indent: 0;
}
/*版权声明*/
p.copyright {
    margin: 1.25em 0 0;
    font-family: "楷体";
    font-size: .9em;
    line-height: 1.6em;
    color: #28363E;
    text-indent: 0;
}
```

版权页背景图，可以发挥自己的想象力，在网络上挑选合适的图片，在

Ps中处理成我们需要的图片，然后导入到ePub中。背景图宽度一般建议在720px以上。

你可以下载我修改的字体样式表「Kindle & DuoKan font-family.css」，然后将字体样式复制粘贴到样式表中。原样式表由 @阜東子安 提供，感谢！

这样封面和版权页就做好了，你可以导入到手机的多看阅读器上看看效果，如果细节上还有问题，可以再次修改。

12.2.6 辅文修改

在这本书，辅文指内容和作者简介、编者献辞、编辑委员会、文前彩页、版本记录等内容，这一节就讲下如何处理这些内容。

内容和作者简介

首先讲下「内容简介」和「作者简介」，为了修饰标题，我们可以在标题前添加一个小图标。比如在「内容简介」前添加一个展开的书籍图标，「作者简介」前添加一个头像图标，本书作者性别是女的，就可以选择一个女性头像图标。

图标的选择建议使用块状的，风格尽量一致，颜色尽量与标题颜色一样。比如如果使用拟物化的就都用拟物化图标，如果是剪影的就都是剪影图标。然后图标的下载我推荐Iconfont，你可以在搜索框中输入关键字，然后选择合适的图标。

照例先给内容分配类，HTML代码如下：

```
<p class="Introduction"><img alt="book" src="../Images/book.png"/>内容简介</p>
<p class="bodycontent">我国是世界文明古国之一，...我国古代文化增添了绚丽的光彩。</p>
<p class="Introduction"><img alt="author" src="../Images/author.png"/>作者简介</p>
<p class="bodycontent">戚志芬，女，山东威海人。...主编《外文有关美国书目》。</p>
```

CSS代码：

```
/*内容和作者简介*/
p.Introduction {
    margin: 1em 0 .5em 0;
    padding-bottom: .25em;
    font-family: "小标宋";
    line-height: 1.5em;
```

```
    color: #a52a2a;
    border-bottom: 1px solid #a52a2a;
}
/*内容和作者简介小图标*/
p.Introduction img {
    height: 1.15em;
    vertical-align: middle;
    margin-right: .35em;
}
```

内容和作者简介的正文，就默认使用正文样式，暂时先不修改。

编辑委员会

编辑委员会的排版我们就尽量还原纸书的排版，排版手段就选择用表格来实现。HTML代码如下：

```
<body class="Contributors-bg">
  <p class="series-book">《中国文化史知识丛书》</p>
  <p class="editors">编辑委员会名单</p>
  <table class="Contributors">
    <tbody>
      <tr>
        <th>主　　编</th>
        <td>任继愈</td>
        <td> </td>
        <td> </td>
      </tr>
      <tr>
        <th><span class="justify">副 主 编<a></a></span></th>
        <td>汤一介</td>
        <td>焦树安</td>
        <td>张明华</td>
      </tr>
      <tr>
        <th colspan="4">编　　委<span style="font-size: .875em;">（按姓名汉语拼音音序排列）</span></th>
      </tr>
      <tr>
        <td> </td>
        <td>戴念祖</td>
        <td>郭齐家</td>
        <td>葛剑雄</td>
      </tr>
      <tr>
        <td> </td>
        <td>焦树安</td>
```

```
            <td>金宜久</td>
            <td>卢海燕</td>
          </tr>
          ...
          <tr>
            <th>常务编委</th>
            <td>任继愈</td>
            <td>焦树安</td>
            <td>张明华</td>
          </tr>
          <tr>
            <td> </td>
            <td>张国风</td>
            <td>郑殿华</td>
            <td> </td>
          </tr>
        </tbody>
    </table>
</body>
```

CSS代码如下：

```
/*编委会背景*/
body.Contributors-bg {
    background-image: url("../Images/Contributors.png");
    background-size: cover;
    background-repeat: no-repeat;
    bacckground-position: center center;
}
/*《中国文化史知识丛书》*/
p.series-book {
    margin: 2em 0 0.5em;
    font-family: "准圆";
    font-size: 1.15em;
    text-indent: 0;
    text-align: center;
}
/*编辑委员会名单*/
p.editors {
    margin: 0 0 2em;
    font-family: "楷体";
    font-size: 1.35em;
    text-indent: 0;
    text-align: center;
}
/*编辑委员会人员*/
table.Contributors {
```

```
    margin: 0 auto;
}
table.Contributors th {
    padding: .25em;
    font-family: "宋体";
    font-size: .875em;
    text-align: left;
    font-weight: normal;
}
table.Contributors td {
    padding: .25em;
    font-family: "仿宋";
    font-size: .875em;
    text-align: left;
}
span.justify {
    display: inline-block;
    text-align: justify;
    width: 4em;
}
span.justify > a {
    display: inline-block;
    width: 100%;
}/*「编委会」文本为三个字，需要对齐，此代码可在Lithium和Sigil中显示完美*/
```

当前HTML文件的页面背景我使用了一张图片，背景图可以从网络中选择，尺寸推荐为1920×1080，单位像素。选择的图片，最好是与原书一样或接近的。

文前彩页

使用多看阅读的交互图代码，插入文前彩页，见10.4.2节。这里不再叙述。

可针对细长的图片单独设置图像宽度，以达到显示美观的目的，即在一屏之内可以完整显示图片和图说。这个就需要在阅读器上检查，然后在Sigil中修改了。一本排版精美的电子书，就是这样反复修改完美的。

编者献辞

先给标题添加类，代码如下：

```
<h1 class="Foreword">编者献辞</h1>
```

然后参考纸书书写CSS，代码如下：

```
/*编者献辞*/
h1.Foreword {
    margin: 30% 0 1.5em;
    font-family: "准圆";
    font-size: 1.35em;
    text-align: center;
    line-height: 1em;
    color: #a52a2a;
}
```

一般我会在HTML文件的第一段内容之上空30%的边距。

然后是正文，HTML代码如下：

```
<p class="bodycontent-fangsong">中国是世界文明古国之一。...一脉相承，历久而弥新。</p>
...
```

编者献辞的正文和正文样式一样，区别在字体的选用：

```
/*仿宋正文*/
p.bodycontent-fangsong {
    margin: .5em 0;
    font-family: "仿宋";
    font-size: 1em;
    line-height: 1.5em;
    text-indent: 2em;
    text-align: justify;
    text-justify: inter-ideograph;
    word-break: break-all;
}
```

这个正文代码，可以从10.3.1节中复制。

编者献辞的最后是落款，参考纸书，添加HTML代码：

```
<p class="inscribe">中国文化史知识丛书编辑委员会<br/><span>一九九六年四月</span></p>
```

双行或以上的落款，建议使用`<br/>`换行，再给这一行添加适当的行高。然后添加CSS：

```
/*落款*/
p.inscribe {
    margin-top: 2em;
    font-family: "楷体";
    text-indent: 0;
```

```
    text-align: right;
    line-height: 1.35em;
}
p.inscribe > span {
    font-family: "宋体";
    font-size: .875em;
}
```

CSS选择器知识见3.3节。

12.2.7　正文部分

正文样式的代码见10.3节，复制粘贴进去。然后使用查找和替换的方式在HTML文件中添加类：

```
<p class="bodycontent">...</p>
```

语法如下：

```
查找：<p>
替换：<p class="bodycontent">
```

Preface的第一段内容添加代码如下：

```
<p class="bodycontent" style="margin-top: 30%;">我国是个历史悠久的文明古国...成了我国的宝贵文化遗产。</p>
```

一级标题下二级标题前的内容，也可以上空30%。

首字下沉

纸书的版式中，有首字下沉效果。见3.4.11节，添加相关代码。

纸书中的首字非常大，我们在ePub中倒不用做那么大，这里的首字大小是正文的2倍。添加首字下沉的段落，建议不要设置首行缩进，即当前段落应用无缩进正文样式（复制正文CSS，修改`text-indent`声明）。

使用正则表达式批量替换，语法如下：

```
查找：<body>\s+<p class="bodycontent">([\x{4e00}-\x{9fa5}])
替换：<body><p class="bodycontent-noindent" style="margin-top:
30%;"><span class="first-word">\1</span>
```

替换完之后，检查首字是标点符号或者是数字的，手动添加。因为上面的正则表达式会遗漏这些，当然你也可以完善这个正则表达式，以匹配更多。

一级标题

参考纸书的版式，书写篇章页的代码。在写代码之前，先收集相应的素材，然后确定HTML代码的结构，最后写CSS。下载的素材，格式如果是`.EPS`，需要用到Adobe Illustrator软件打开，然后配合Photoshop软件，这是都是很基础的操作，这里不再叙述。

HTML代码如下：

```
<div class="chapter">
  <img alt="QingLong" src="../Images/QingLong.png"/>
  <p>一</p>
</div>
<div style="clear: right;"></div>
<h1 class="title-1">类   书</h1>
<p class="book-pinyin">ZHONG GUO<br/> WEN HUA SHI<br/> ZHI SHI<br/> CONG SHU</p>
```

然后是CSS代码：

```
/*篇章页*/
div.chapter {
    width: 6em;
    margin: -0.25em 0 2em;
    float: right;
    padding-top: 1em;
    background-color: rgba(165,42,42,0.5);
    text-align: center;
}
div.chapter > img {
    width: 5em;
    margin: 0 0 5em;
}
div.chapter > p {
    margin: 0 0 0.5em;
    font-family: "黑体";
    font-size: 2em;
    color: #f7f7f7;
}
/*一级标题*/
h1.title-1 {
    margin: 0 0 2.5em;
    font-family: "黑体";
    font-size: 2em;
    font-weight: normal;
    text-align: right;
    color: #a52a2a;
```

```
}
p.book-pinyin {
    margin: 0;
    font-family: "准圆";
    line-height: 1.5em;
    text-align: right;
}
```

然后使用正则表达式批量替换，语法如下：

```
查找：<h1>([\x{4e00}-\x{9fa5}]) (.*?)</h1>
替换：<div class="chapter"><img alt="QingLong" src="../Images/QingLong.png"/><p>\1</p></div><div style="clear: right;"></div><h1 class="title-1">\2</h1><p class="book-pinyin">ZHONG GUO<br/> WEN HUA SHI<br/> ZHI SHI<br/> CONG SHU</p>
```

完成后，可以格式化下代码。同时检查下每个篇章页的实际显示效果，进行细节调整。新用户使用手机首次打开多看阅读器时，字号是+4，所以要保证篇章页在这个字号下没有换页。

二级标题

一级标题做好之后，开始制作二级标题。参考纸书的样式，书写代码。HTML代码如下：

```
<h2 class="title-2">1 什么是类书？</h2>
```

CSS代码：

```
/*二级标题*/
h2.title-2 {
    margin: 30% 0 1.5em;
    font-family: "楷体";
    font-size: 1.35em;
    font-weight: normal;
    line-height: 1.35em;
    color: #a52a2a;
}
```

然后使用正则表达式批量替换，语法如下：

```
查找：<h2>(.*?)</h2>
替换：<h2 class="title-2">\1</h2>
```

完成后，检查下，有显示不完美的地方，进行细节的修改。

其他正文

通览纸书的PDF文件，遇到需要制作样式的在电子书中添加进去。比如在一章的4节（P27），有小标题需要加粗，这里用黑体代替或者可以用小标宋。HTML代码如下：

```
<h3 class="title-3">（1）曹魏和南北朝的重要类书</h3>
```

CSS代码如下：

```
/*三级标题*/
h3.title-3 {
    margin: .875em 0 .875em 2em;
    font-family: "黑体";
    font-weight: normal;
    line-height: 1em;
    font-size: 1em;
}
```

由于多看对标题标签限制了缩进，所以使用外边距的方式添加缩进。然后使用正则表达式针对这种规律的文本，批量替换。语法如下：

```
查找：<p class="bodycontent">（(\d{1,})）(.*?)</p>
替换：<h3 class="title-3">（\1）\2</h3>
```

P28页开始有行头加粗的样式，HTML代码如下：

```
<span class="bold">a 类书的始祖——《皇览》</span>
```

CSS代码如下：

```
/*行内加粗*/
span.bold {
    font-family: "小标宋";
}
```

因为TXT文本是经过初步整理的，所以可以使用正则表达式去查找。

然后使用正则表达式批量查找替换，语法如下：

```
查找：<p class="bodycontent">([a-z]) (.*?)
替换：<p class="bodycontent"><span class="bold">\1 \2</span>
```

再继续浏览PDF文件，如二章2节的3级标题参考上文添加小标题样式。

注音

接着是拼音了，书宋的拼音字符没有经过优化，显示效果并不理想。我

们可以单独给注音部分添加字体样式。HTML代码如下：

```
〔<span class="pinyin">dì 第</span>〕
```

CSS代码如下：

```
/*拼音*/
span.pinyin {
    font-family: "DK-SYMBOL","宋体";
}
```

DK-SYMBOL只是符号字体，不包含中文，在实际显示中会使用系统的字体显示中文，所以后面再加一个书宋字体。然后使用正则表达式批量查找替换，语法如下：

```
查找：〔(.*?)〕
替换：〔<span class="pinyin">\1</span>〕
```

然后是注释部分内容的修改。

12.2.8　注释内容

因为注释的特殊性，所以单独拎出来讲。可以使用一位网友　@王萌萌开发的Sigil插件，具体使用教程参考8.5.5节。

不过这个文本经过我的整理，注释都经过重新编号，所以用不用插件都可以。在使用正则表达式格式化注释之前，先更改注释的计数序号类型。这里使用搜索模板的方式，你可以下载我写好的圆圈序号（①）改方括号（[1]）形式搜索模板，然后批量替换。具体教程见5.3.17节。

在替换后，注意正文中是否存在与注释编号一样的计数，如果有需要在全部替换之后再修改回来。

因为之前通过批量的方式给所有的<p>都加了正文样式，所以我们需要使用正则表达式，给注释内容删去正文样式。语法如下：

```
查找：<p class="bodycontent">\[(\d+)\]
替换：<p>[\1]
```

HTML代码

由于我使用的这个文本，本身注释序号就已经编好序号，所以我先用正则的方式批量修改文中文下注释。首先，先替换文下注释内容（文下注释格式见10.6.3节），正则表达式语法：

```
查找：<p>\[(\d+)\](.*?)</p>
替换：<li class="duokan-footnote-item" id="footnote-\1"><p>\2</p></
li>
```

然后添加首尾列表代码
。

```
查找：<li class="duokan-footnote-item" id="footnote-1">
替换：<ol class="duokan-footnote-content"><li class="duokan-
footnote-item" id="footnote-1">
查找：</li>\s+</body>
替换：</li></ol></body>
```

接着是文中注释编号，正则表达式语法如下：

```
查找：\[(\d+)\]
替换：<a class="duokan-footnote" href="#footnote-\1"><img alt=""
src=" ../Images/note.png"/></a>
```

完成以后，就可以增加注释图标和CSS了。

CSS代码

首先是注释图标，可以使用10.6.3一节演示的注释图标。CSS代码：

```
/*多看注释*/
a.duokan-footnote img {
    width: .875em;
    vertical-align: middle;
}
ol.duokan-footnote-content {
    margin-top: 1.5em;
    border-top: 1px solid #333;
}
li.duokan-footnote-item {
    margin: .35em 0;
}
li.duokan-footnote-item p {
    margin: 0;
    font-family: "宋体";
    font-size: .875em;
    line-height: 1.35em;
    text-indent: 0;
    text-align: justify;
    text-justify: inter-ideograph;
    word-break: break-all;
}
```

这样，注释内容就修改好了。然后删除不需要的文本，如「注释：」。

12.2.9　最后工作

完成以上的工作之后，一本电子书就差不多做完了。不过我们还需要继续做一些工作：元数据、质检、版本记录等。

元数据

现在开始编辑电子书的元数据，见1.2.2和5.3.8二节。元数据信息模板如下：

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>中国的类书、政书和丛书</dc:title>
  <dc:relation>中国文化史知识丛书·002</dc:relation>
  <dc:creator opf:role="aut">戚志芬</dc:creator>
  <dc:publisher>商务印书馆</dc:publisher>
  <dc:source>CNKI</dc:source>
  <dc:description>我国是世界文明古国之一，历史文化典籍，浩瀚烟海，为帮助青少年了解这些珍贵的文化遗产，本书选择其中的类书、政书和丛书，分为三部分扼要介绍；第一部分按时代顺序，介绍历代重要类书的作用和史的发展道路；第二、三部分，分别介绍政书、丛书的类型和历史。这三者是我国丰富的文化遗产中的重要部分，是人类智慧的结晶，共同为我国古代文化增添了绚丽的光彩。</dc:description>
  <dc:identifier opf:scheme="ISBN">7-100-02103-0</dc:identifier>
  <dc:date opf:event="publication">1996-12</dc:date>
  <dc:date opf:event="creation">2017-09-16</dc:date>
  <dc:subject>百科全书-丛书-典章制度-古籍-基本知识-中国-古代</dc:subject>
  <dc:type clc="Z12;Z22">中国丛书；中国百科全书、类书</dc:type>
  <dc:language>zh-CN</dc:language>
</metadata>
```

自定义属性clc指的是中国图书馆分类法，可以在纸书中的版权页中查询。

质检

完成元数据的添加之后，见5.4.19节，检查ePub的错误。

参考章节：

8.2　pagina-EPUB-Checker——验证ePub
8.5.4　使用EpubCheck插件验证ePub文件

质检没有问题之后，可以导入到手机中，查看实际的预览效果。有问题的地方，再调整细节。比如注释是否可以弹注、篇章页有没有换页等。

版本记录

记录版本号、创建时间等信息。当版本有变动时，可描述变动的内容，给读者以参考。

建议

做书的目的是看书，所以当做完一本书后，最好全书看一遍。不光完成看书的任务，同时也检查了一遍全书。这样更容易发现制作中遗漏的问题，没有阅读过自己做的书，怎么能做出读者满意的电子书！

下载我做好的成品——中国文化史知识丛书·002·中国的类书、政书和丛书。

12.3 Word文档制作ePub

本次演示的案例制作仍是针对多看阅读。

Word文档是办公软件Microsoft Office Word或WPS创建的一种文件，这种格式的电子书与TXT文档的区别是，Word文档可以记录图书的版式信息。不少图书即是用Word排版的，所以研究如何把Word文档转制成ePub格式是非常有必要的。

在8.5.1一节中讲了使用Sigil插件转换Word文档，这种方式的缺点也很明显：

1. 行内样式，如楷体、仿宋等无法保留；
2. 区别于正文的段落样式无法保留；

这给带来不少问题，所以需要另外的方法去处理这种情况。Word文档制作ePub的思路有2种：一种是使用插件导入到Sigil；另一种是把Word文档存为网页文件，删除不需要的标签信息，保留有用的样式信息，把原始素材处理到相对干净为止。一般另存为出来的网页文件都比较大，所以第一步是先格式化标题，然后拆分文件生成目录，接着再一步步的制作。

12.3.1 整理代码

本次教程使用的原素材是经过@观者排版的《中国历史故事：秦 西汉》，已联系本人，获得转载许可。

使用Microsoft Office Word软件打开下载好的Word文档，依次点击主菜单「文件→另存为」，选择保存的文件夹，保存类型为「筛选过的网页」：

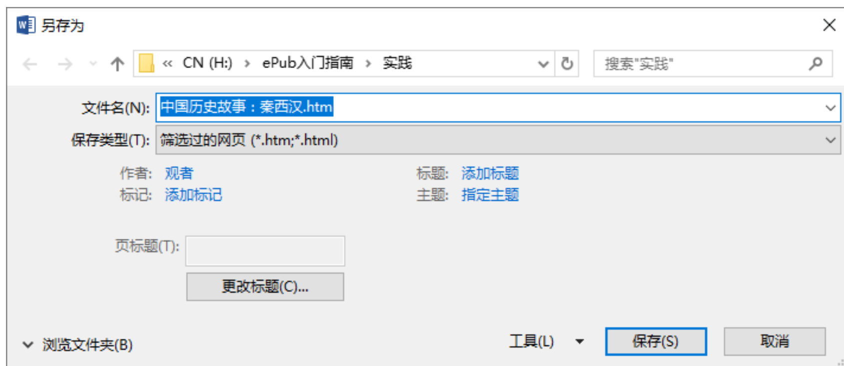


图12-14　Word另存为HTML文件文件

然后点击「保存」按钮。在弹出窗口中，点击「是」按钮，如果弹出下面的提示：

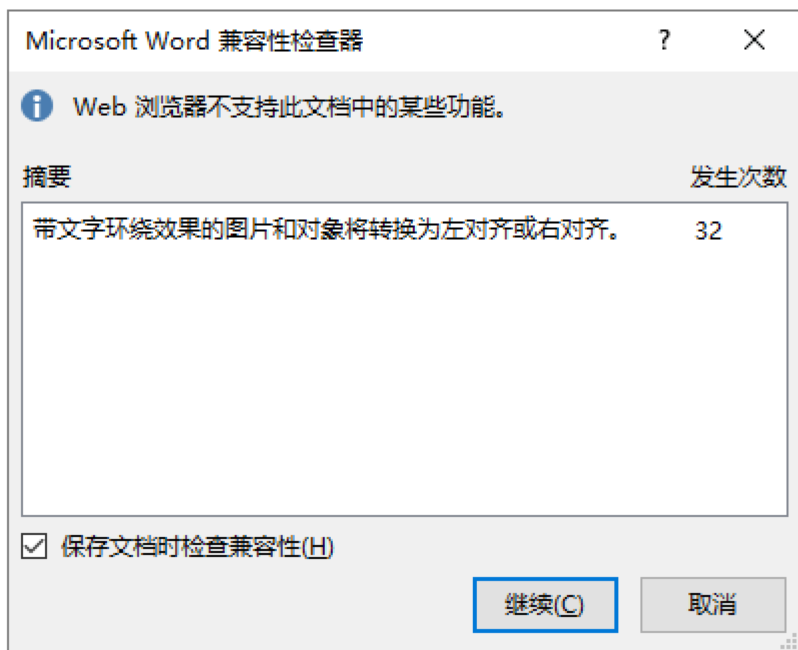


图12-15　兼容性检查

点击「继续」按钮，稍等片刻，网页就转换好了。

打开Sigil软件，使用「打开」功能打开这些网页，见5.3.3节，这里不再叙述。在处理之前，先保存一次。

htm文件顶部的`<style>`标签内容有大量的注释，处理建议是删除。删除后，保存文件。

如果是这种棕色的样式代码，我们需要先给它合并断行。

```
<p class="MsoNormal" style='text-indent:21.0pt;line-height:16.3pt;layout-grid-mode:
both;punctuation-wrap:hanging'><span lang="EN-US"> </span></p>

<p class="MsoNormal" style='margin-left:22.35pt;text-indent:21.0pt'><span lang="EN-US"> </span></p>

<p class="MsoNormal" style='margin-left:22.35pt;text-indent:21.0pt'><img width="124" height="65" src="../Images/image002.png"
align="left" hspace="12"/></p>

<p class="MsoNormal" style='margin-left:22.35pt;text-indent:21.0pt'><span lang="EN-US"> </span></p>

<p class="MsoNormal" style='margin-left:22.35pt;text-indent:21.0pt'><span lang="EN-US"> </span></p>

<p class="MsoNormal" style='margin-top:6.0pt;margin-right:0cm;margin-bottom:0cm;
margin-left:22.4pt;margin-bottom:.0001pt;text-indent:21.0pt'><span lang="EN-US"> </span></p>

<p class="MsoNormal" align="right" style='margin-right:15.75pt;text-align:right;
text-indent:0cm;line-height:normal;word-break:break-all'><span style='font-size:22.0pt;font-family:方正仿宋_GBK;color:red;letter-
spacing:2.0pt'>中国历史故</span><span style='font-size:22.0pt;font-family:方正仿宋_GBK;color:red'>事</span></p>
```

图12-16　合并代码非标准换行

处理办法是直接复制相关代码，查找替换。内联样式换行的比较多，请耐心替换。可以通过直接查找：和；　，然后把空格替换掉。

处理内联样式

使用一个插件，帮我们把内联样式转换成外部式。插件名RemoveInLineStyles，下载之后，在Sigil中安装。这里不再叙述，见8.5一节。

在使用插件处理内联样式之前，建议你保存一份副本。依次点击「插件→编辑→RemoveInLineStyles」，稍等片刻，等待插件运行完成。

删除类

插件运行完成后，会在Styles文件夹中创建一个CSS文件。这么多的选择，有些是不需要的，建议删除选择器和HTML文件中的相关标签。比如选择器rule_68和rule_28，这是Word为了做出两端对齐而增加的代码，在这里我们就不需要。首先现在CSS文件中删除这两个选择器。回到代码视图，使用正则表达式批量查找替换，语法如下：

```
查找：<span class="rule_28">(.*?)</span>
替换：\1
```

正则表达式写完之后，先测试一下，无误后，全文替换。同类的样式还有rule_46、rule_56、rule_78等等，删除这些无用的类时，最好到Word文档中确认一下。

这一步非常费时，同时要注意不能误删。所以，在涉及全文批量操作时，移动确认无误后再替换。

代码清理的差不多时，就可以开始下一步骤了。

12.3.2　生成目录

观察Word的目录，以一级标题为单位拆分正文，标题和内容在同一个网

页中。使用下面的正则表达式语法，进行查找替换：

```
查找：<div class="WordSection(\d+)">\s+<h1>
替换：<hr class="sigil_split_marker" /><div class="WordSection
\1"><h1>
```

拆分标记添加好之后，确认无误然后执行拆分。拆分完成后，对HTML文件进行命名。见12.2.4节命名。新增如下：

- 扉页/书名页 Titlepage
- 贡献者 Ccontributors

其中「目次」这一页，需要删除，不用保留。电子书不需要这种目录。命名完成以后，我们还需要修改下网页文件的格式类型，把htm改为xhtml。步骤：选择需要修改的网页文件，点击鼠标右键，选择「重命名」选项，在弹出对话框中输入如下：

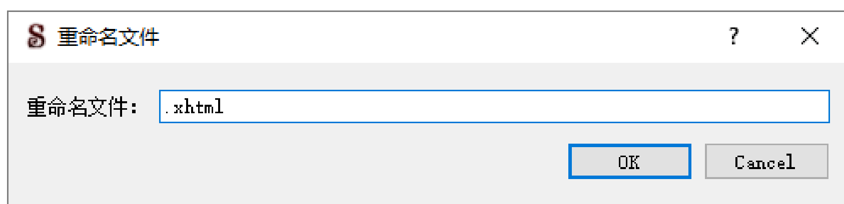


图12-17 修改内容文档扩展名

点击「OK」按钮，执行操作。完成后如下（部分）：



图12-18 修改好的内容文档

在生成目录之前，我们先给标题标签修改一下。现在的标题代码如下：

```
<h1><a name="_Toc487126580"><span class="rule_43">一、奇货可居
</span></a></h1>
```

需要做的是删除a、span元素，给h1添加类。那么正则表达式语法如下：

```
查找：<h1><a name="(.*?)"><span class="rule_(\d+)">(.*?)</span></
a></h1>
```

```
替换：<h1 class="title-1">\3</h1>
```

替换完成后保存文件，见12.2.4节处理目录，这里不再叙述。目录完成后如下：

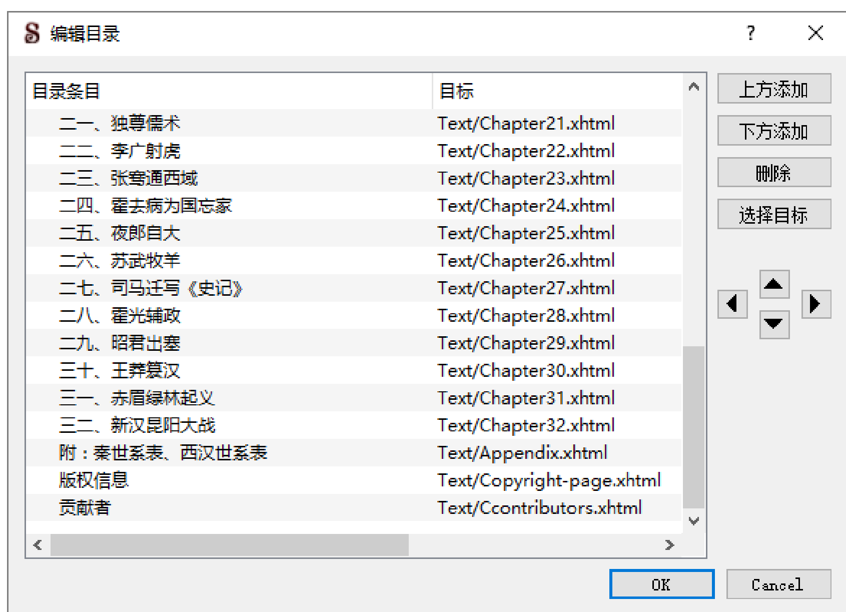


图12-19　编辑目录

本书经过@观者校对，他是一位很厉害的校对人员，所以这里就不再列出版本信息。

一定要记住，在拆分完网页之后，就先做好目录。这对于后面的修改，有非常大的帮助。也切记，目录编辑好之后，如无必要就不再动目录。

12.3.3　辅文排版

在12.3.1节中只是清理了一些明显的无用代码，但实际上从Word文档转成的网页文件，有大量的垃圾代码存在。有时虽然不影响排版，但是稍微有代码洁癖的就受不了。所以为了观感，我们还需要再次清理下代码。同时也开始对正文进行排版，排版的过程中可以参考原始素材。

封面

仍然从第一个网页——封面开始，这里我们不使用Sigil的「添加封面」功能，而是直接从上一个实践案例中复制代码，粘贴进来。需要处理如下：

1. HTML代码替换，添加封面语义（注意代码中的图片宽高数值）；

2. 封面文件命名更改，同时修改封面尺寸，添加封面图片语义；
3. OPF文件增加全屏属性。

上面步骤请参考相关章节，自行查找修改方法，这里不再叙述。

Word另存出来的网页，图片尺寸会比实际的图片尺寸要小很多，所以需要再Word文档中把图片另存为，然后再导入ePub中。

扉页

头部文件中删除无关meta元素，如：

```
<meta content="text/html; charset=gb2312" http-equiv="Content-Type"/>
<meta name="Generator" content="Microsoft Word 15 (filtered)"/>
```

使用查找替换的方式批量删除。替换时，注释查找替换的模式，是正则表达式还是正常。

title元素中添加当前页面的名称，如扉页。

body元素的开始，删除无关属性。语法如下：

```
查找：<body link="#0563C1" class="rule_1" vlink="#954F72" lang="ZH-CN">
替换：<body>
```

然后是类似`<div class="WordSection1">`的代码，为了避免误删，先使用正则表达式批量删除，语法如下：

```
查找：<div class="WordSection\d+">
替换为空
```

然后依次点击「工具→HTML重新格式化→Mend and Prettify All HTML Files」。这样做的好处是，Sigil可以自己判断哪些`</div>`是需要删除的。

到了这一步就涉及到CSS的书写了，所以我们先给样式表重命名。然后载入我提供的字体样式表。

HTML代码如下：

```
<p class="center"><img style="width: 7em;" src="../Images/image002.png"/></p>
<p class="book-title">中国历史故事</p>
<p class="book-subtitle">秦　西汉</p>
<p class="book-author">朱仲玉编写</p>
<p class="center"><img style="width: 8em;" src="../Images/
```

```
publisher.png"/></p>
<p class="title-sign">书名题字：<span>茅　盾</span></p>
```

然后参考Word文档的版式书写样式。

CSS代码：

```
/*元素居中*/
p.center {
    text-align: center;
    margin: .5em 0;
    line-height: 1em;
    font-family: "宋体";
}
/*扉页中的书名*/
p.book-title {
    margin: 1em 0 0.5em;
    font-family: "仿宋";
    font-size: 1.8em;
    text-align: center;
    color: #f75555;
    font-weight: bold;
}
p.book-subtitle {
    margin: 0;
    font-family: "黑体";
    text-align: center;
}
/*扉页中的作者*/
p.book-author {
    margin: 1em 0 50%;
    font-family: "仿宋";
    text-align: center;
}
/*题字*/
p.title-sign {
    margin: 2em 0;
    font-family: "仿宋";
    font-size: .875em;
    text-align: center;
}
p.title-sign span {
    font-family: "黑体";
}
```

完成后，导入ePub到手机中，查看效果，没有问题后开始编辑下一个HTML文件。

内容说明

内容说明的样式比较简单，标题格式化一下就行。HTML代码：

```
<p class="Introduction">内容说明</p>
<p class="bodycontent">我国历史悠久，流传着许多有...代的《中国历史故事》。</p>
<p class="bodycontent">这一册是秦和西汉部分，共有三...借鉴的地方。</p>
```

CSS代码：

```
/*内容说明*/
p.Introduction {
    margin: 3em 0 1em;
    font-family: "准圆";
    font-size: 1.25em;
    text-align: center;
}
```

附

附录部分需要删除不需要的标签和属性，使用下面的正则表达式在当前文件中处理。语法如下：

```
查找：<span (.*?)>(.*?)</span>
替换：\2
查找：<p (.*?)>(.*?)</p>
替换：\2
```

给table元素添加类，参考Word文档排版。样式见10.8一节内容排版。这里使用三线表的样式，同时根据原Word的排版再原样式上再增加单元格左右边框。HTML代码：

```
<p class="table-title">秦世系表<br/><span>（公元前246—公元前207年）</span></p>
<table class="three-line-table">
...
</table>
<p class="bodycontent">注：秦始皇二十六年开始称皇帝，二十六年以前称秦王</p>
<p class="table-title" style="page-break-before: always;">西汉世系表<br/><span>（公元前206年—公元9年）</span></p>
<table class="three-line-table">
...
</table>
```

可以自己定义分页行为，把第二个表格分开显示。表格代码如上面提到的三线表代码一样，这里不再贴出。CSS代码：

```
/*表格名*/
p.table-title {
    margin: 1.5em 0 0.5em;
    font-family: "仿宋";
    font-size: 1.25em;
    font-weight: bold;
    text-align: center;
    line-height: 1.15em;
    color: #a52a2a;
}
p.table-title span {
    font-family: "宋体";
    font-weight: normal;
    font-size: .75em;
}
```

给但特定单元格增加左右边框，建议使用nth-child选择器，不过使用这种选择器的缺点是，多看阅读器并不支持，所以只能在Sigil中可以看到效果。CSS代码如下：

```
table.three-line-table th:nth-child(2),table.three-line-table td:nth-child(2) {
    border-right: 1px solid #444;
    border-left: 1px solid #444;
}
```

版权信息

版权信息完全按照Word文档的样式排版，需要注意的是把字号调到+4，让所有信息都在一屏之内显示。HTML代码：

```
<p class="fangsong">封面设计：潘絮兹</p>
<p class="fangsong" style="margin-bottom: 25%;">插　　图：贺友直</p>
<p class="book-subtitle">中国历史故事（秦　西汉）</p>
<p class="center">朱 仲 玉编写</p>
<p class="center">*</p>
<div class="publisher">
  <p><img style="height: 1.25em;vertical-align: middle;" alt="publisher" src="../Images/publisher.png"/> 出版</p>
  <p>中国青年出版社印刷厂印刷</p>
  <p>新华书店北京发行所发行　各地新华书店经售</p>
</div>
```

```html
<p class="center">*</p>
<div class="publisher2">
  <p>787×1092 1/32  5.5印张  88千字</p>
  <p>1979年12月北京第1版  1979年12月北京第1次印刷</p>
  <p>印数1 - 300,000册  定价0.32元</p>
</div>
<p class="right"><span class="isbn">书号R11056·54</span><br/>定价    0.32元</p>
```

CSS代码：

```css
/*居中的仿宋*/
p.fangsong {
    text-align: center;
    font-family: "仿宋";
    margin: .35em;
}
div.publisher p {
    margin: .5em 0;
    line-height: 1em;
    text-align: center;
    font-family: "宋体";
    font-size: .85em;
}
div.publisher2 p {
    margin: .5em 0;
    line-height: 1em;
    text-align: center;
    font-family: "宋体";
    font-size: .7em;
}
/*居右*/
p.right {
    text-align: right;
    line-height: 1.5em;
    font-family: "宋体";
    font-size: .875em;
}
/*书号*/
span.isbn {
    padding-bottom: 1px;
    border-bottom: 1px solid #2e74b5;
}
```

书号的下划线，可以在Word中截图，在Ps中新建画布粘贴进去，然后放大画布，使用吸管工具吸取颜色。

最后的贡献者没什么好说的，需要注意的是微信和支付宝这两张图片的大小。

12.3.4 正文排版

下面是关于正文的排版，这部分内容有正文、拼音、插图、注释等。

注音

正文中存在不少注音，见12.2.7节，添加相关代码，这里不再叙述。附录的表格中三个字有注音，原书是行间注，为了看起来整齐，我使用ruby元素，对拼音进行了排版。见2.2.11节，添加HTML代码及样式。

```
/*拼音*/
span.pinyin,ruby rt {
    font-family: "DK-SYMBOL","宋体";
}
```

正文

正文样式仍然采用上一个实践的正文样式，直接复制相关代码，粘贴进CSS文件中。然后HTML文件的代码，使用查找替换的方式替换类。

插图之后的段落是无缩进正文，与正文的区别是首行缩进为零。

标题

这本书只有一级标题，之前已经添加了类，现在给这类写样式，主要参考Word文档。每篇文章之后有一个小图案，我们可以提前，放到页面开始的位置。不过部分章节后面没有，所以我的建议是删除这个图片。CSS代码：

```
/*一级标题*/
h1.title-1 {
    margin: 20% 0 1.5em;
    font-family: "仿宋";
    font-size: 1.45em;
    text-align: center;
    color: #a52a2a;
}
```

然后使用正则表达式把标题内容添加到title元素中，见4.3节。

插图

参考排版样例，处理插图。上文中讲过，另存为网页中的图片尺寸变小

了，所以还需要使用Word文档中的图片。如果是本书演示的这种插图，还需要做抠图处理。同时背景透明的图片不建议使用交互代码。

本次演示的案例插图抠图步骤：

1. 在Ps中打开图片；
2. 依次点击「图像→模式→灰度」；
3. 依次点击「图像→模式→RGB颜色」；
4. 解锁图层；
5. 使用魔棒工具，选择白色背景，按Delete删除背景；
6. 保存图片。

插图都是黑白的，所以可以使用图片优化工具，压缩下图片大小。这里使用的是PPDuck软件，免费版一次可以处理10张图片，如果超出10张，重启软件就行。

步骤：先把ePub中的图片另存出来，然后使用皮皮鸭压缩，完成以后再导入ePub中。

压缩前建议先把插图备份！

其他

最后，在Word中浏览全文，检查其他需要制作的样式。比如P64的诗歌，我们可以做成居中的格式：

```
<div class="poemtry">
  <p>力拔山兮气盖世，时不利兮骓不逝。</p>
  <p>骓不逝兮可奈何，虞兮虞兮奈若何！</p>
</div>
```

CSS代码：

```
/*诗歌居中*/
div.poemtry {
    margin: .875em 0;
}
div.poemtry p {
    margin: .35em 0;
    text-align: center;
    font-family: "仿宋";
    line-height: 1em;
}
```

在P105也有一首诗需要排版。

在P100原书注音错误，@观者指出了这里可以使用多看弹注方式标注出

来。示例：

```
<a class="duokan-footnote"><img alt="原书：剽悍〔biǎo hàn〕" src="../Images/note.png"/></a>
```

然后给注释图片设置下大小和垂直对齐方式。

其他还有P104、P146、P170等。

最后使用Sigil的删除未使用样式功能，查找最后没有找出来的样式，没用的话就直接删掉。

然后还是上一个章节提到的流程，验证→修改→试读。

元数据

参考上一个实践，自行添加。

下载我制作的成品——中国历史故事：秦 西汉。

12.3.5 制作总结

Word文档转制ePub的主要问题是生成的网页，垃圾代码非常多，所以清理起来非常的费时间。如果一本s是非常大、样式又超级复杂的书，做起来就需要非常多的时间。

本次教程一般流程如下：

1. Word文档另存为筛选过的网页；
2. 使用Sigil打开这些网页；
3. 合并内联样式的换行；
4. 使用RemoveInLineStyles插件，抽取类；
5. 删除无用且样式重复的类；
6. 格式化标题标签；
7. 确定拆分规则，拆分网页，然后重命名网页；
8. 生成目录及编辑目录；
9. 辅文排版，样式参考原书并适当调整；
10. 正文排版，插图、引文、诗歌等，图片重命名；
11. 替换插图并决定是否做抠图、压缩处理；
12. 浏览Word文档，查看是否有遗漏的样式未添加；
13. 清理未使用的媒体文件、样式；
14. 添加图书的元数据信息；
15. 验证ePub文件并修改，试读；
16. 成品文件重命名，发布。

制作过程中切记随时保存，重大修改前保存副本！

12.4　图像PDF转TXT格式文本

在1.4一节介绍过图像PDF的电子书知识，这类电子书需要用OCR技术，识别出文本，见8.1.2节。至于矢量或双层PDF文件，则可以使用无极天软件直接输出为ePub，然后在此基础上进行精排版。无极天软件为商业软件，使用需购买，操作方式与掌阅的iRemaker类似。

一般图像PDF的制作流程如下：

1. 图书预处理，图像二值化；
2. 使用不同的专业工具OCR出两份文本（ABBYY、汉王）；
3. 使用BCompare软件比对两份文本，综合正确率；
4. 人工读校并整理文本；
5. 合并非标准换行（TextForever、宏脚本）；
6. 开始制作成ePub（实践一）；
7. 试读过程中二校；
8. 修改二校中发现的错误。

其中部分教程取自 @心情复杂的颖小喵 写的《bc大法 V1.0》教程，感谢！

本章不讲制作ePub的具体步骤，因为在上两个实践中已经讲过了。我们现在所需要的只是一份TXT格式的文本。

12.4.1　图片处理

需要做OCR的文件一般有PDF、UVZ、PDG等格式，我们需要统一处理成JPG或TIFF格式。然后二值化图像，提高OCR的正确率。

导出图片

对于PDF格式的电子书，可以使用Acrobat的导出功能直接把PDF文件的每一页导出为JPG格式的图像文件。下面是具体步骤：

使用书名新建一个文件夹，再把PDF文件放在这个文件夹中。使用Acrobat软件打开待处理的PDF文件，然后依次点击「文件→导出到→图

像→JPEG」，弹出如下对话框：

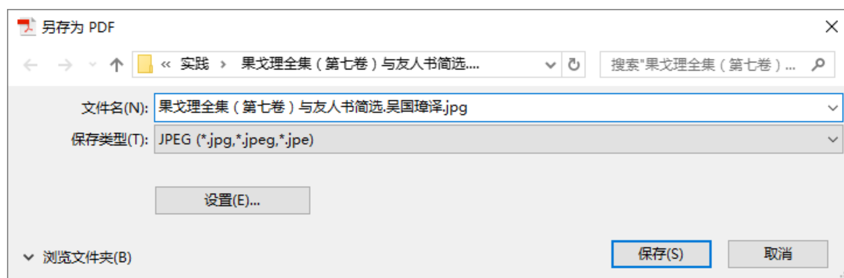


图12-20　导出PDF页面为图像

点击「设置」按钮。设置参数如下：



图12-21　设置图像参数

完成后点击「确定」按钮，在「另存为PDF」对话框中点击「保存」按钮，开始保存图像文件。保存的时候最好不要再做其他的事，以免电脑崩溃卡顿，需要重新保存。图像保存以后，可以删除不需要的页面，比如封面、扉页、版权页、目录页等。

还有一种是UVZ格式的电子书，这种电子书和PDF类似。如果想要把它转换成JPG格式，需要用到Pdg2Pic软件。软件所在目录：老马的原创空间＞原创软件。

备份UVZ文件，在副本上修改文件类型为zip，然后解压到当前文件夹中。解压出来的文件即是PDG格式，这是只有超星阅读器可以阅读的文件格式。这里我以《唐五代两宋词选释》为例。

打开Pdg2Pic软件，在「1、选择需转换的PDG文件所在文件夹」中选择解压的文件夹。然后勾选「选择存放转换出来的图像文件的文件夹」。

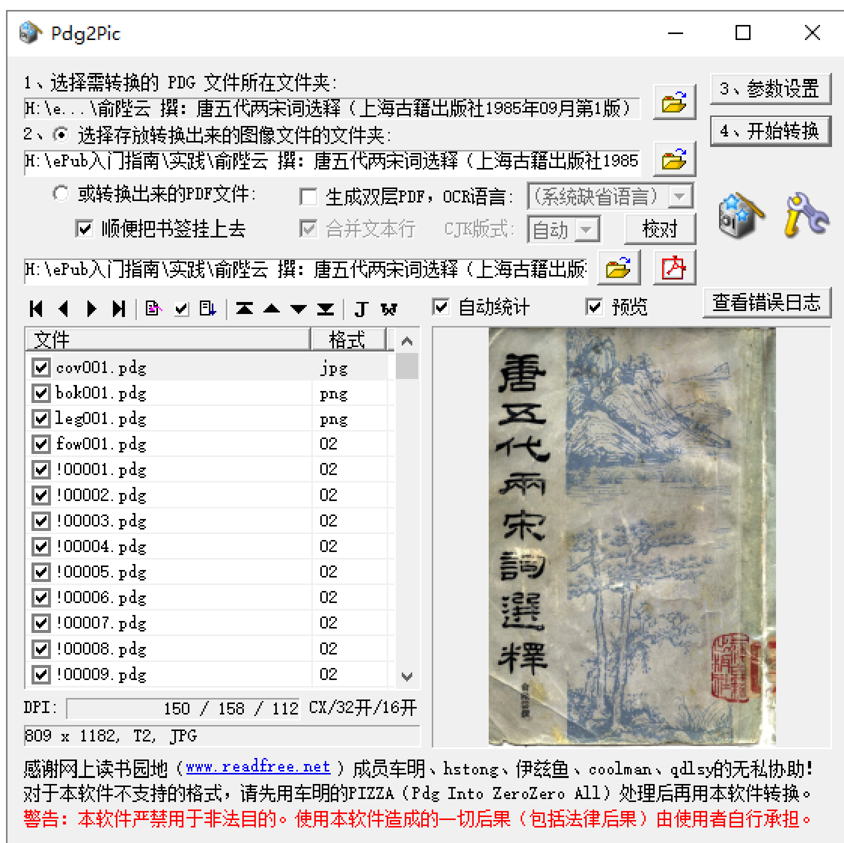


图12-22　PDG文件转成图像文件

点击「3、参数设置」，取消「重新编号转换出来的图像文件」的勾选，以免重新编号后，再转回PDG时还要改文件名。另外因为最后还要转回PDG，FreePic2Pdf的接口文件在里面只会捣乱，所以在「参数设置」中取消它。在Pdg2Pic的「参数设置」中如果选择了「转换结束后删除原始PDG文件」，或者将图像文件设置到了一个新的文件夹，那么转换后目标文件夹中就只有图像文件。否则转换出来的图像文件与原始PDG文件混合存放在同一个文件夹下。参数设置如下：

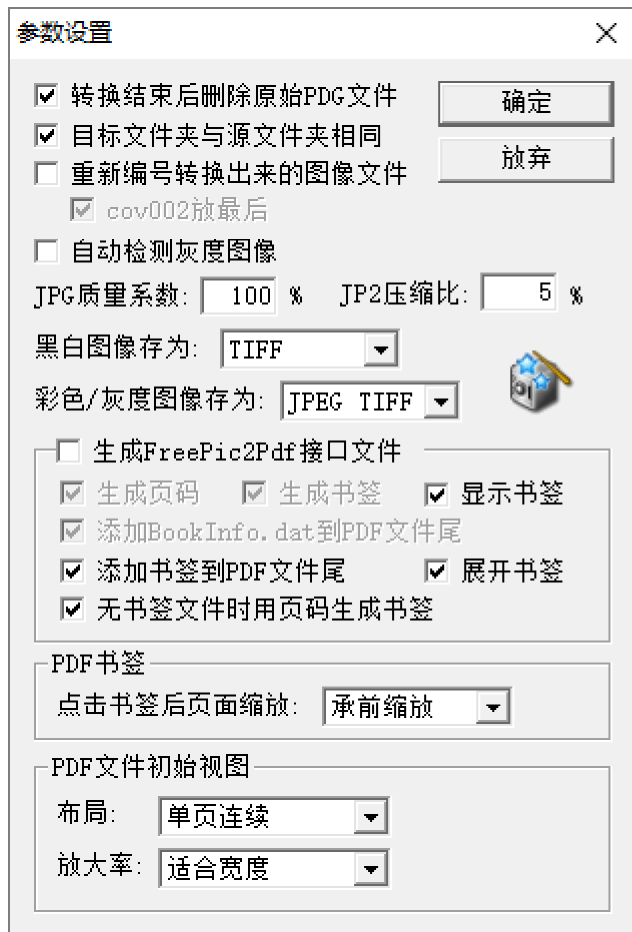


图12-23 图像参数设置

参数设置完成以后，点击「确定」按钮保存并退出该对话框。然后点击「4、开始转换」按钮，开始转换。

转换完成以后，如果你想要保存在保存一份PDF文件，可以在Pdg2Pic主界面中勾选「或转换出来的PDF文件」并勾选「顺便把书签挂上去」选项。然后点击「4、开始转换」按钮，开始转换。

二值化

下面的教程摘自老马的《ComicEnhancerPro系列教程》，感谢大佬！

由于原始扫描图像左、右页的边距不同，为了在切边时能更准确地切边，需要把转换出来的图像按照左、右页分开。另外封面不需要处理，因此也需要单独存放到其他地方。

新建一个文本文档，命名为「111_222a」，然后使用记事本程序打开

它。复制下面的代码，粘贴进文本文档中。

```
md 111
md 222
for %%a in (1 3 5 7 9) do move *%%a.* 111
for %%a in (0 2 4 6 8) do move *%%a.* 222
```

然后保存文本文档，修改文件类型为bat。

使用的时候把`111_222a.bat`文件复制到需要处理的图像文件所在文件夹，双击运行，然后就看到黑色的DOS命令行窗口在滚动，搞定后自动关闭。完成后如下：

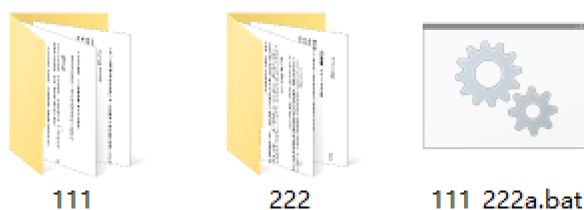


图12-24　奇偶页分类

其中111文件夹存放奇数页，222文件夹存放偶数页。

下载ComicEnhancerPro软件，软件所在目录：老马的原创空间＞原创软件。

启动CEP，或关闭CEP中所有打开的窗口后点「恢复」按钮复位所有参数。

打开奇数页文件夹中的文件。

点「缩放」页签，拉到200%。

点「其它」页签，点「色彩」按钮，「色彩数」选「纯黑白」，「阀值算法」选Otsu，「去斑直径」选8。

点「切边」页签，点「设置按钮」，打开「切边」对话框后先把它点到最大，这样切边的时候看得更清楚。

在「切边」对话框中，点翻页按钮，定位到一个满是文字的页面：

1. 因为文字是横排，所以纠斜选择「自动_横排」。
2. 用鼠标拉动右侧预览框中的黑色控制点，选择大致的内容框，具体值参见下面的附图。
3. 勾选「手动选择范围内自动选择」，「忽略斑点直径」增加到10。缺省的「不计黑边」、「斑点补偿」保持勾选状态。

4. 点左侧下方的翻页键，把所有页面全部看一遍，看有没有出现误切的，如果有则调整手工选择范围，或改变上一步中的选项。
5. 页面大小选择「内容框大小」或「保持不变，内容框外填白」（推荐），此时预览界面会有图中的绿色线（也就是处理后留下来的实际文本区域），红色外框是指所需文本的最大边框，如果有页眉页脚需要去掉，将红色外框调整至合适位置后，浏览其他页面，检测是否全部合适。
6. 因为前面步骤2中选择了放大200%，因此把DPI从300改成600。
7. 看一下界面上显示的版心尺寸和内容框尺寸，实际内容框的宽度比版心宽度小了一点，但长度超了一点，估计是因为页码的关系。因此「内容框排版」选择：「水平」选「居中」，实现内容框水平居中。「垂直」选「靠上」，这样至少保持顶部平齐。当然其中版权页的位置会出现较大变化，最后再处理。当内容框大于版心时溢出，这样对内容框就不会进行裁剪。因为实际显示的当前内容框像素尺寸已经大于版心像素尺寸，因此「当内容框小于版心时」选项没啥意义，就保持缺省的「填白」。

选完以后的「切边」对话框如下图所示，点「确定」按钮退出该对话框，并使参数发挥实际作用。

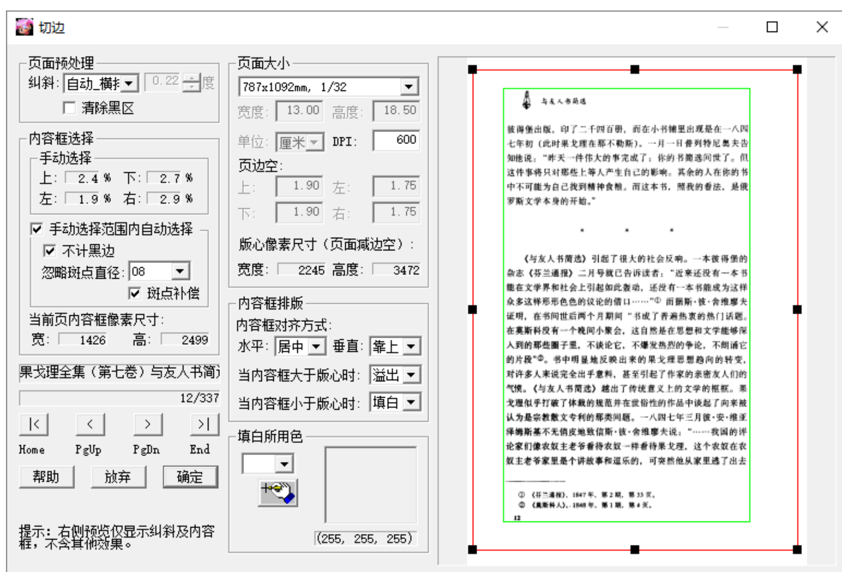


图12-25　页面切边

看一下结果图像，内容框居中了，画面也干净了，如果文字边缘锯齿太严重，点「其它」页签，点「调节」按钮，「高斯模糊半径」选0.7，「锐化百分比」选35。以上参数以实际显示为准。

点工具条上的「批量转换」按钮，「存放新文件的文件夹」选择上级文件夹，点「全部转换」开始转换。

处理完奇数页后，不要急着退出CEP，而是关闭奇数页窗口，直接打开偶数页文件，这样就可以重复利用奇数页的参数设置。

需要修改的是手动切边范围。

批量转换，「存放新文件的文件夹」与奇数页选择相同的文件夹，转换后自然就合在一起了。把封面再移回来，一本书就齐了。

由于前面用Pdg2Pic转换的时候去掉了「重新编号转换出来的图像文件」选项，因此事情很简单，简单的DOS批处理：

```
ren *.tif *.pdg
ren *.jpg *.pdg
ren *.gif *.pdg
ren *.bmp *.pdg
```

参考上面的步骤新建一个文本文档，修改文件类型为bat，命名为「Pic_2_Pdg」。

就可以把处理后的图像转换成能够用CV、UV直接看、能够用Pdg2Pic直接转换成PDF的PDG文件。使用的时候把Pic_2_Pdg.bat文件复制到需要更名的文件所在文件夹，双击运行即可。

12.4.2　开始OCR图书

处理完图像以后就可以开始OCR了，这里选择ABBYY和汉王两款软件。使用两个软件分别OCR同一本书，得到两份文本。

ABBYY

ABBYY是一家提供PC和移动设备光学字符识别、文档识别软件和计算机辅助语言学习软件的国际软件公司。ABBYY的大多数产品，如ABBYY FineReader，意在简化从纸质文档到数字数据的转换过程。ABBYY也提供基于语言的软件产品和服务。

本次演示使用的软件版本是ABBYY FineReader 12 Corporate Portable版，请自行搜索版本名，这里不提供下载链接。

在上一节已经把《果戈理全集 第七卷》的页面转成了图片，我就以这本书为例。

启动软件后，依次点击「文件→打开PDF文件或图像」，找到图像所在文件夹，选择需要识别的页面，点击「打开」按钮，然后等待软件读取完毕。

> （此处以 @捕鱼 的方法为例）为了提高识别的正确率，语言一定要选择正确（比如：简体中文＋英语），但默认的简体中文貌似是CJK字典，会识别出一些异体字，可以参考Abbyy Finereader ocr减少出错率的小技巧教程进行替换。

识别完成以后，点击保存图标右侧的▾，选择「另存为PDF文档」：

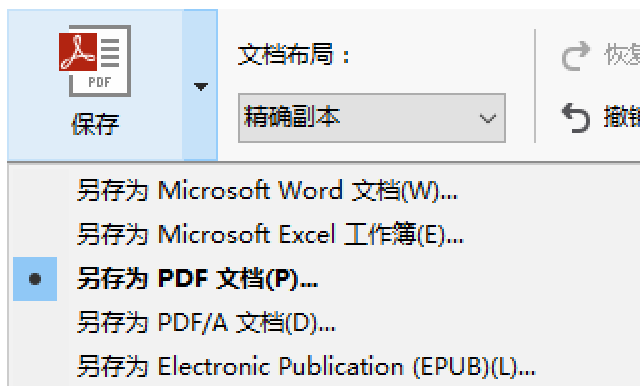


图12-26　另存为PDF文档

点击「格式选项」按钮，保存模式设置为「页面图像下方的文本」，也就是双层可搜索PDF，便于读校时查找文本。点击「确定」按钮，保存设置并退出当前对话框。重命名PDF文件，然后点击「保存」按钮，保存PDF。

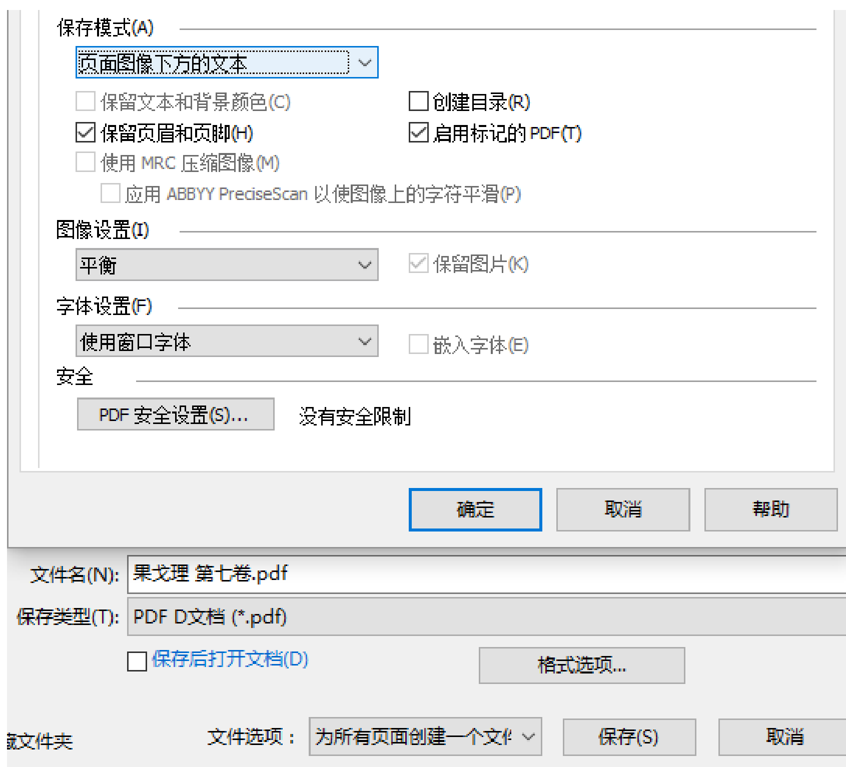


图12-27　PDF保存模式

完成后，回到ABBYY主界面，再次点击保存图标右侧的▾，选择「保存为其他格式」选项，保存类型中选择「文本(*.txt)」，点击「格式选项」按钮，文本设置中勾选「保留换行」和「使用空白行作为段落分隔符」。点击「确定」按钮，保存设置并退出当前对话框。

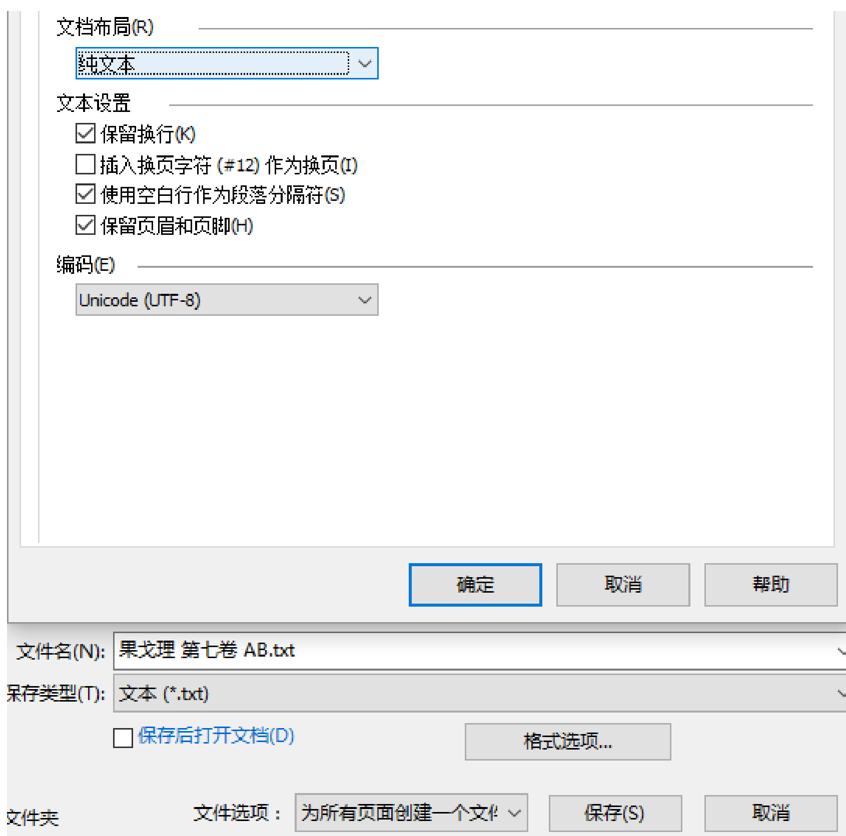


图12-28　文档布局

文件命名为「书名 AB.txt」，点击「保存」按钮，这就完成在ABBYY软件中的识别工作。然后可以关闭ABBYY软件了。

汉王

汉王科技股份有限公司成立于1998年，是全球文字识别技术与智能交互产品引领者，多年来，通过不断自主创新，在手写识别、光学字符识别（OCR）、笔迹输入等领域拥有多项具有自主知识产权的核心技术，综合技术水平在国内外均处于领先地位，手写汉字识别获得国家科技进步一等奖，OCR获得国家科技进步二等奖。

这里使用的是Hwpdfocr_V8.1.4.16版本。

依次点击「文件→打开图像」，在弹出窗口中选择图像所在文件夹，然后选择需要识别的页面，点击「打开」按钮。汉王打开图像时，似乎有数量限制，所以需要多次选择一定数量的图像，直到所有图像都在汉王中打开。

依次点击「识别→选择全部文件」，然后再依次点击「识别→开始识别」，等待软件OCR完成。

识别完成以后，全选左侧的图片列表，依次点击「输出→到指定格式文件」，选择文件要保存的文件夹，保存类型为文本文件，文件命名为「书名　HW.txt」，然后点击「保存」按钮。等待软件保存完毕。完成后，就可以关闭汉王软件了。

全部完成后，我们就获得了两份文本，一份「果戈理 第七卷 AB.txt」，一份「果戈理 第七卷 HW.txt」。

12.4.3　处理文本错误

在比对之前先使用宏脚本处理下文本常见错误。这里使用的是排版专用宏 20170525.jsee，你可以下载它，然后安装到EmEditor中。教程见7.5一节。

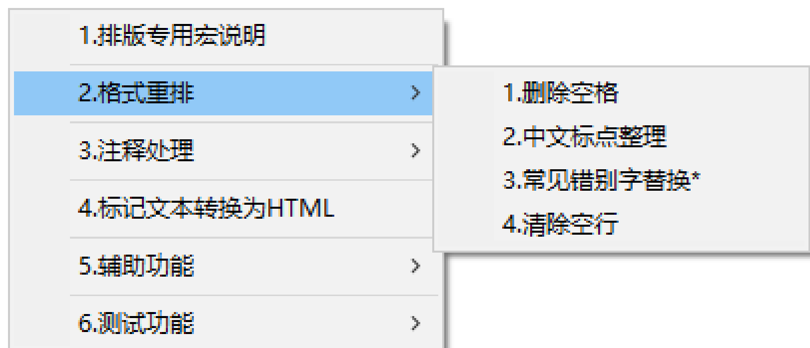


图12-29　文本格式处理

此脚本可以去除空格、标点整理、删除空行等。

依次执行删除空格（需要多次运行）、中文标点整理、常见错别字替换*。整理完成后，保存文件。

> OCR出来的文本不要合并断行。

12.4.4　比对文本

Bc是Beyond compare 4软件的简称，作用在于比对文本的差异，然后纠错。经实践证明比（大部分人的）肉眼一校靠谱（且速度更快）。此方法并非自创，出处已不可考，再次感谢诸位前辈！

下载好软件和配置文件之后，依次点击「工具→导入设置」，在弹出对话框中选择配置文件「BCSettings.bcpkg」，然后点击「打开」按钮。

在「导入设置」对话框中点击「下一步」按钮，默认全部勾选，继续点击「下一步」按钮，选择「文本比较」和「文本合并」，接着就一路默认。

选择「果戈理 第七卷 AB.txt」和「果戈理 第七卷 HW.txt」，直接拖动文件到「文本比较」图标上松手。

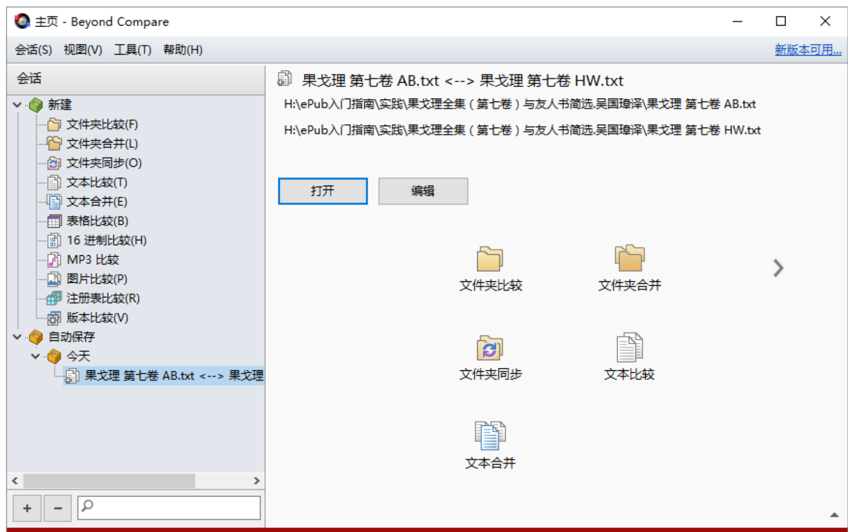


图12-30　Beyond compare操作界面

完成后如下（上下视图），也可以更改为左右视图：



图12-31　文本比对

若要更改为左右视图，依次点击「视图→边并排布局」即可。

左侧放ABBYY识别出来的文本，红色部分表示差异。如果肉眼看觉得是一样的，但显示为红色，放大字号再观察。右侧放汉王识别出来的文本。

紧接着下面就是每一行的对比区域，光标放在这里再按快捷键，可以只复制该行到另一侧，而不用特别的快捷键。如果看不懂这句话，多实践几次就明白了。

导入配置后，更改文本到另一侧的默认快捷键是Ctrl＋U（右侧/下面的文本对，复制到左侧/上面。这里的U可以理解为Up）和Ctrl＋D（Down，向下）。查找下一个差异是Ctrl＋N（Next，下一个），上一个是Ctrl＋P（Previous，前一个）。撤销修改的默认快捷键是Ctrl＋Z，这与大部分软件一样。

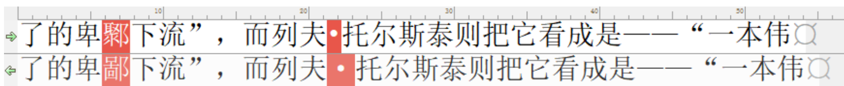


图12-32　文本比对缩略窗口

如果文本上某个字词，两个软件都识别错误，自行在左侧、右侧等区域替换。另外，该软件只能复制一行到另一边，所以如果一行出现两个错误，一个ABBYY对，一个汉王对，那只能手动修改一处，然后再复制到另一边了。

有时一个字（尤其是一些生僻字）会被两个软件都识别成错误，并且识别为同一个，BC只能看差异，所以是显示不出来的，只能靠读校的时

候，对照PDF改正。

> 如果左右两侧的文本不能进行编辑，将光标放到任何一个区域内，点击鼠标右键，选择「完整编辑」即可。

下图这种情况，上面的符号代表此处在文本内有个空白行，下面在此处没有空白行。此时需要搜索双层PDF，查询该处是否需要分段，再进行修改（比如原书这里分段了，就留下换行符，以便后期合并的时候处理）。



图12-33　空行符号

BC的时候可以只改一侧的文档，即以左侧/上方的文档为基准，如果左侧/上方是对的，则不进行任何修改，最终关闭软件的时候会提示你要不要存到右侧/下方，点「是」即可。

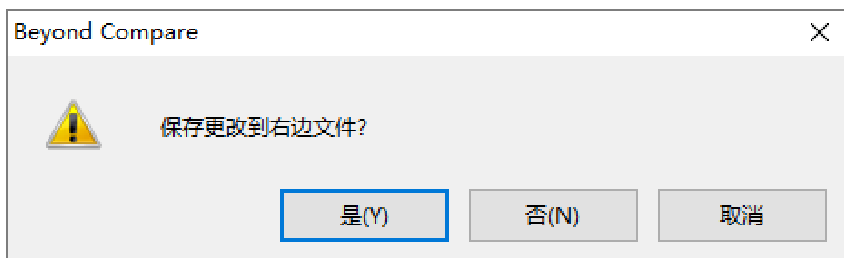


图12-34　更改提示

或者直接另存为一份文件——注意存的是你修改的左侧/上方的文件。这样可以少改很多出，但如果不熟练的话有可能会遗漏，所以……自己看着办吧。

12.4.5　合并段落

BC完以后的文本是按照PDF文件换行的，与正确的换行不同，所以需要合并非标准换行。此处采用 @捕鱼 的方法，用TxtForever合并段落。也可以使用别的方法。



图12-35　TxtForever操作界面

设置参数如上，点击「选择」，在打开对话框中选择BC过的文本，然后点击「开始合并」。

此处是根据前面ABBYY识别后段落间留空行的设置来合并的，然而机器毕竟是机器，如果遇到PDF里面一排字最后一个符号是句号、叹号、省略号、问号之类的标点，很可能会被ABBYY认为该处换段，而在后面加了空行，此时合并之后也是错误的分段。所以需要依照PDF再检查一遍段落是否分对了。然后按照9.13节整理TXT文本。

之后就可以开始排版了，见12.2节。排版完成后，读校——也就是二次校对。以读为主，发现错误时记录笔记。全书看完以后，在电脑上对照PDF修改，见7.4节。BC工作做好的话，其实读校错误已经非常少了，所以并不麻烦。但机器不是万能的，要保证文本质量，一定要进行二次校对！

12.5 AZW3/MOBI制作ePub

随着电子书的数量增多，我们制作的资源有时也会是成品格式，还有就是AZW3、MOBI。如果是亚马逊的电子书格式，想要在Sigil中编辑时，需要使用KindleImport插件。插件的安装见8.5一节。

对于这类格式的电子书，它的本质仍然是ePub，不过是半成品。需要再精加工，针对阅读器做特别适配、修改样式等。所以这与TXT文本制作ePub的后半段类似，主要流程如下：

1. AZW3/MOBI转成ePub格式；
2. 删除不需要的文件：网页、图片、样式表等；
3. 重命名文件；
4. 拆分文件并格式化标题；
5. 添加辅助页：封面、版权等；
6. 编辑目录；
7. 设计样式；
8. 添加图书元数据；
9. 测试、修改。

修改这类格式的主要工具是正则表达式，一般成品在样式方面上大都已经统一了，用正则表达式修改会比较快。

12.5.1 Kindle电子书转ePub

下载好的文件需要先进行格式转换，以便可以在Sigil中可以编辑。无论是AZW3还是MOBI格式，都可以用KindleImport插件转换成ePub格式。转换后的文件，删除我们排版时不需要用的文件，比如网页、图片、样式表等。

另外如果是AZW3/MOBI格式，安装好KindleImport插件后，依次点击「插件→输入→KindleImport」，Sigil会提示你插件正在运行：

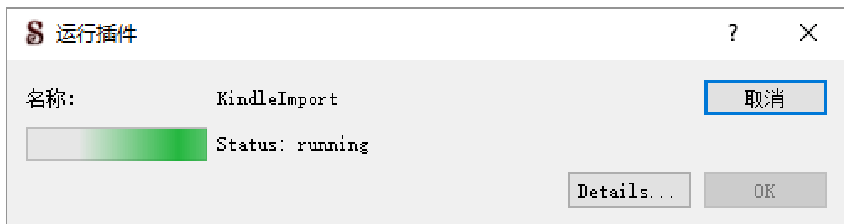


图12-36　KindleImport正在运行...

过几秒后，会弹出一个选择Kindle电子书的窗口，选择你要转换的电子书文件：

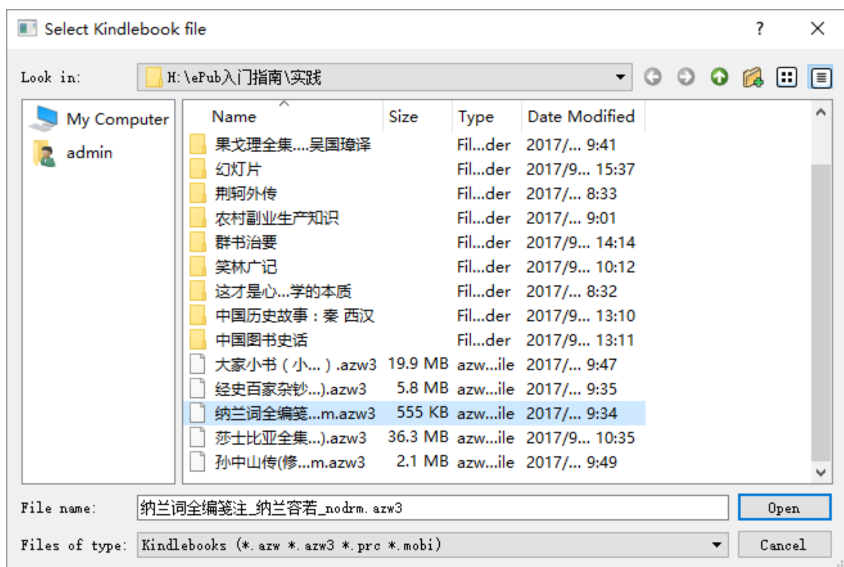


图12-37　选择要转换的azw3文件

选择好之后，点击「Open」按钮，插件会转换Kindle文件。这个过程视待转换文件的大小而定，可能几秒，也可能几分钟。转换完成后，会弹出如下：

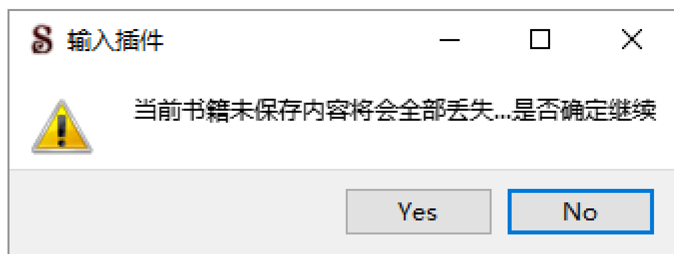


图12-38　输入提示

如果你的Sigil没有在编辑别的书籍，选择「Yes」，Sigil会加载转换的电

子书，然后保存时重命名ePub文件即可。

如果你的Sigil正在编辑书籍，选择「No」，退出转换操作。在Sigil中保存你在编辑的电子书，然后重新打开Sigil，再输入Kindle电子书。

12.5.2　整理文件及确定目录

本次演示用的文件为《Python教程-廖雪峰》，源文件格式是ePub，适配平台是多看阅读器。

下载好上面的案例文件，用Sigil打开它。首先看到的是网页文件使用了中文命名，目录结构没有源网站清楚。首先就针对这两个问题修改，参考电子书作者提供的教程地址，然后修改ePub。

Python简介 1
安装Python 1
Python解释器
第一个Python程序 1
使用文本编辑器 2
Python代码运行助手 2
输入和输出 2
Python基础 1

图12-39　epub素材目录

数字1即是一级标题，2是二级标题。对中ePub中的网页修改标题层级，见5.4.8节。

不过我在编辑时，可以明显感觉到卡顿，然后Sigil就崩溃了。排查之后，发现是部分网页中嵌入了外链视频。它们是：

- 2安装Python.html
- 4第一个Python程序.html
- 5使用文本编辑器.html
- 17定义函数.html

4个网页，5个视频。先提取视频地址，在视频插入出标记符号，然后删除视频代码。使用下载工具（迅雷等）下载视频或到教程网页中视频另存为，因为在多看中在线视频无法播放。视频地址：

```
http://asklxf.coding.me/liaoxuefeng/v/python/install-py.mp4
http://asklxf.coding.me/liaoxuefeng/v/python/start-py.mp4
http://asklxf.coding.me/liaoxuefeng/v/python/first-py-code.mp4
http://asklxf.coding.me/liaoxuefeng/v/python/run-hello.py.mp4
http://michaelliao.gitcafe.io/video/py/import-abstest.mp4
```

处理好之后，参考网站标题层级进行修改。子级在另外一个网页中的，合并相应网页。做完以后，生成目录，在「选择要包括的标题」中，设置为只包括一级和二级。

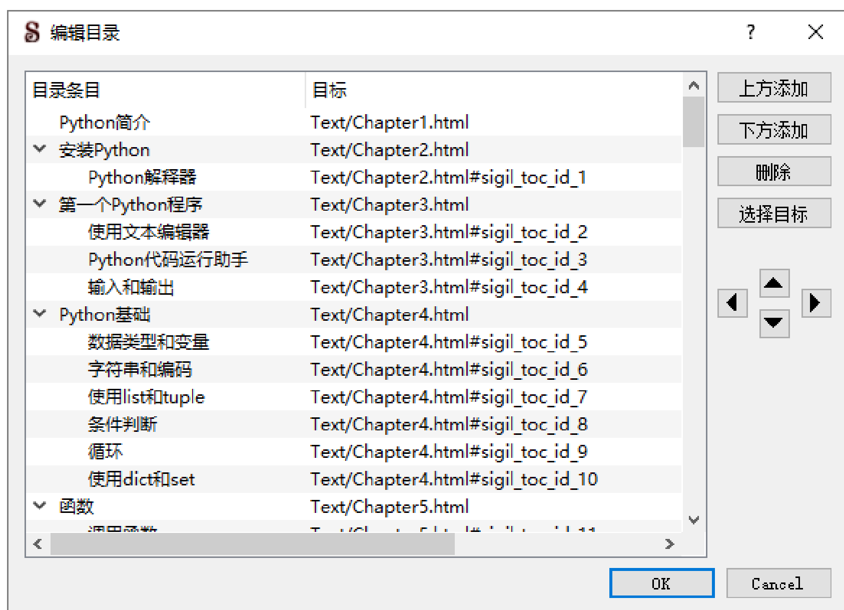


图12-40　编辑目录

清理标题

然后开始整理小标题，如果网页中的内容本身是标题标签包裹的，在生成目录时设置为隐藏，则会生成一个无用的类 `class="sigil_not_in_toc"`，根据这个规则，把这些标题标签都统一格式化成一种样式。原标题代码如下：

```
<h3 class="sigil_not_in_toc"><a name="#-E5-AE-89-E8-A3-85Python-3-5"></a>安装Python 3.5</h3>
<h3 class="sigil_not_in_toc"><a name="#-E5-9C-A8Mac-E4-B8-8A-E5-AE-89-E8-A3-85Python"></a>在Mac上安装Python</h3>
...
```

其中a元素我们不需要，先使用正则删除。语法如下：

```
查找：<a name="#(.*?)"></a>
替换为空
```

确认无误后，全文替换。然后这些小标题添加一个自定义的类`class="little-title"`，在Styles文件夹中新建一个样式表文件，命名为*Stylesheet.css*。全书网页链接样式表，在样式表中增加如下代码：

```
p.little-title {
    margin: .875em 0 0;
    font-family: "黑体";
    color: #a52a2a;
}
```

正则表达式如法如下：

```
查找：<h3 class="sigil_not_in_toc">(.*?)</h3>
替换：<p class="little-title">\1</p>
```

检查有没有漏掉的小标题，比如h4/h5等。为了区别小标题样式，可以再增加一个样式，类名为sublittle-title。



图12-41　标题排版

上面little-title为棕色，sublittle-title为黑色。

12.5.3　删除无用的代码

整理好文件及目录之后就可以开始清理无用的代码了，在案例中每个网页有下面的代码包裹正文：

```
<body>
  <div class="x-content" style="width:100%">
  ...
  </div>
</body>
```

这里我们需要删除div元素，步骤是使用查找替换的方式，删除`<div`

`class="x-content"  style="width:100%">`，然后再用清理代码的方式，清理网页。依次点击「HTML重新格式化→Mend and Prettify All HTML Files」，这样就清理了div元素。

接着是阅读量：

```
<div class="x-wiki-info">
   <span>Reads: 2168606</span>
</div>
<hr/>
```

这种信息也不需要，使用正则表达式删除。语法如下：

```
查找：<div class="x-wiki-info">\s+<span>Reads: (\d+)</span>\s+</div>
\s+<hr/>
替换为空
```

确认无误后，全文替换。后面还有这种，通过阅读类之后，可以判断是无用的代码：

```
<div class="x-wiki-content x-main-content">
...
</div>
```

参考上面的内容删除此类代码。在「代码视图」下，浏览全文查看有无需要删除的代码。

12.5.4 精加工

精加工这一步主要是排版标题、图片、代码等内容。

标题

参考教程网页给标题设置样式，尽量简单。HTML代码如下：

```
<h1>Python简介</h1>
```

CSS代码：

```
h1 {
    margin: 30% 0 1.5em;
    padding-bottom: .25em;
    font-family: "黑体";
    font-weight: normal;
    font-size: 1.35em;
```

```
    border-bottom: 1px solid #d3d3d3;
    color: #a64b4b;
}
```

然后是二级标题，CSS代码如下：

```
h2 {
    margin: 1.5em 0 0.5em;
    padding-bottom: .25em;
    font-family: "黑体";
    font-weight: normal;
    font-size: 1.15em;
    border-bottom: 1px dashed #d3d3d3;
    color: #d93636;
}
```

插图

接着是插图，可以看到的问题是，父级元素是<p>标签，图片使用中文命名。这两点都是不推荐的做法，先给图片重命名，然后修改文中代码。打开Images文件夹，选择所有图片，点击鼠标右键，选择「重命名」选项，在弹出对话框中输入Picture，然后点击「OK」按钮，确认操作。

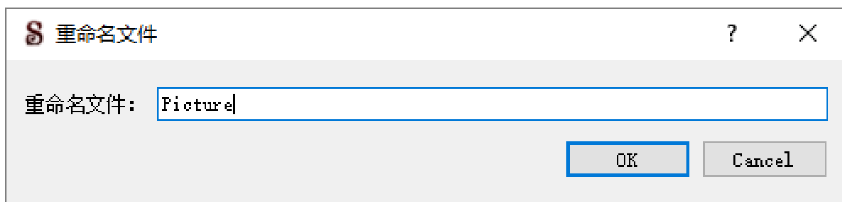


图12-42　重命名图像文件

图片重命名之后，代码如下：

```
<p><img alt="1Python简介_1" src="../Images/Picture31.jpg"/></p>
```

其中alt属性中的内容是图片名称，正则表达式语法如下：

```
查找：<p><img alt="(.*?)" src="../Images/Picture(\d+)\.jpg"/></p>
替换：<div class="picture_box"><img alt="\1" src="../Images/Picture\2.jpg"/></div>
```

确认无误后，全文替换。完成后先清理网页，代码如下：

```
<div class="picture_box">
  <img alt="1Python简介_1" src="../Images/Picture31.jpg"/>
```

```
</div>
```

然在*Stylesheet.css*文件中增加如下CSS样式：

```css
div.picture_box {
    width: 100%;
    text-align: center;
    margin: .5em 0;
}
```

正文

正文样式使用排版案例的中的样式，见10.3节。然后删除首行缩进，调整行间距、段间距。

超链接

默认的超链接是蓝色字体加下划线，这比较丑，我们可以去掉下划线只保留字体颜色，字体颜色取自教程上的颜色。代码：

```css
a {
    color: #0593d3;
    text-decoration: none;
}
```

在Chrome浏览器中，按下快捷键F12进入开发者窗口。



图12-43　超链接颜色

点击红色的数字1处，然后选择要查看的元素对象，即数字2，点击完之

后，在右侧下面显示样式信息，其中红色箭头所指的即是该超链接所使用的颜色值。双击即可以复制该16进制颜色代码，然后粘贴进电子书中。

有些文内的超链接，比如「定义函数」一节中的。检查网页中的教程，查看该超链接的指向位置，然后修改电子书中的超链接指向。

代码

部分代码是用<code>、<pre>标签包裹的，这类内容一般表示的是代码。虽然在Sigil中可以看到字体与正文字体不一样，不过在多看阅读器中是不能区分显示的。给这两种标签增加下面的代码：

```
code {
    padding: 0 4px;
    border: 1px solid #ddd;
    border-radius: 3px;
    background: #fafafa;
    font-size: .875em;
    color: #D05;
    white-space: nowrap;
    font-family: "code";
}
```

上面的代码同样可以使用开发者工具查看到。然后浏览正文，发现下面的代码与教程不一样：

```
<pre><code>『python』 不是内部或外部命令，也不是可运行的程序或批处理文件。 </code></pre>
```

显示如下：

情况二：得到一个错误：

'python' 不是内部或外部命令，也不是可运行的程序或批处理文件。

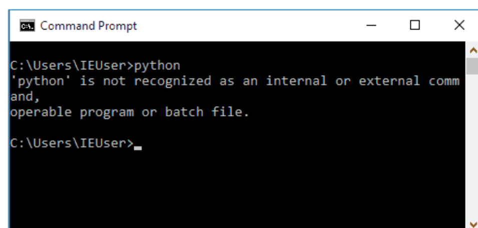


图12-44　代码格式

这显然不是我们需要的样式。增加下面的代码到样式表中：

```
pre {
```

```
    margin: .5em 0 0.5em;
    word-break: break-all;
    word-wrap: break-word;
    white-space: pre-wrap;
    padding: .5em;
    background: #fafafa;
    font-size: .875em;
    font-family: "code";
    color: #444;
    border: 1px solid #ddd;
    border-radius: 3px;
}
```

浏览网页上的教程，部分代码还需要换行，直接从网页上复制代码，粘贴到电子书中。这一步比较费时，请耐心制作，仔细检查。

视频

把已经下载好的视频导入到电子书中，见10.7.2节插入视频。视频封面图可以直接从网页上截图或者再视频播放器中截屏。示例代码：

```
<video class="duokan-video content-matrix" poster="../Images/start-py.jpg">
  <source src="../Video/start-py.mp4" type="video/mp4"/>
  您的浏览器不支持video标签。
</video>
```

列表

部分列表是使用p元素包裹的，使用正则表达式删除p元素：

```
查找：<li><p>(.*?)</p></li>
替换：<li>\1</li>
```

然后增加如下样式：

```
ul,ol {
    margin: .875em 0;
    font-size: .875em;
    font-family: "准圆";
    color: #666;
}
li {
    margin: .35em;
    line-height: 1.35em;
}
```

表格

参考网页中的教程设置表格样式。CSS代码如下：

```
table.table {
    margin: .875em 0;
    width: 100%;
    font-family: "黑体";
    font-size: .875em;
    vertical-align: middle;
}
table.table th {
    padding: .25em;
    text-align: left;
    border-bottom: 1px solid #d3d3d3;
}
table.table td {
    padding: .25em;
    text-align: left;
    border-bottom: 1px solid #d3d3d3;
    color: #444;
}
```

title

见4.3一节，修改网页的title。

12.5.5　最后工作

下面要做的其实是每本书都要做的内容，比如增加封面、填写元数据、质检、常见文本错误检查等。

增加封面

本教程中没有相应的封面，我们可以在网络上随便找一本动物书做参考，然后在Ps里制作一张封面。尺寸见10.5节。相应的Ps教程见6一章。或在在这个网站（http://oreilly-generator.com/）各生成一个封面，缺点是生成的封面分辨率较低。

至于动物书指的是O『reilly出版的计算机相关的图书，一般这种书都会有一个动物图像。本教程的封面就是参考这种风格设计的。

元数据、质检

参考前两章的相关章节自行填写。

常见文本错误

这类指的是无用的空格、多余的代码、非标准换行等等。请自行编写正则表达式查找。

下载我制作的成品——Python教程-廖雪峰。

12.6 常见问题

本节搜录一些常见问题的解决方法，这些问题实际是前文的知识点组合应用。如果写在文中，反而不容易查找，所以另开一节放在12章。

12.6.1 网络小说的制作

网络小说的章节标题格式一般是统一的，这种文本有两种方式去处理。一种是12.2节的处理，在EmEditor中清除行首缩进、段落之间空一行、转成Unicode编码，然后用Sigil直接打开。打开以后，切换到代码视图模式，清理下代码。写正则表达式，批量添加拆分标记，然后执行拆分。关于章节标题的格式，收集如下（默认标题以p元素包裹）：

表29 网文常见标题格式

格式	正则
第一章 城里没有我的家	查找：第([\x{4e00}-\x{9fa5}]{1,})章 (.*?) 替换：第\1章 \2
三十二 陷入危机	查找：([\x{4e00}-\x{9fa5}]{1,}) (.*?) 替换：\1 \2
第1章 报仇何须十年？	查找：第([0-9]{1,})章 (.*?) 替换：第\1章 \2
18 集市卖肉	查找：([0-9]{1,}) (.*?) 替换：\1 \2

上面就是最基本的4种标题格式，还有一些变种。比如「章」变成「节」、「回」、「话」、「卷」等，标题序号即章/节/回等后面的空格，变成冒号（：）、全角空格（　）等。诸如此类，直接修改上面的正则即可。下面提供一个示例：

```
查找：<p>第([\x{4e00}-\x{9fa5}]{1,})章：(.*?)</p>
替换：<hr class="sigil_split_marker" /><h1>第\1章：\2</h1>
```

上面正则的意思是把第x章这一段落，格式化成一级标题，并在一级标题前添加拆分标记。在运行正则表达式之前，先进行「查找和替换」面板的设置，一般在「选项」中勾选「循环匹配」，「模式」选择「正

常」、「当前文件」和「下」。完成设置后，即可运行正则表达式。成功后，执行拆分内容文档操作，然后保存。

另外一种方法就是用老牛的EpubSTAR，智能分章，然后用Sigil再精加工。也可以用掌上书苑的EpubBuilder制作。这两种软件，使用起来都很简单，这里不再讨论。

12.6.2 合并多个ePub电子书

合并多个ePub电子书，主要用的是Sigil的「添加现有文件」功能，步骤如下：

1. 打开电子书1，全部选中Text文件夹中的内容文档，点击鼠标右键，在菜单中选择「另存为」，保存到一个文件夹中。
2. 全部选中Images文件夹中的图片文件，点击鼠标右键，在菜单中选择「另存为」，保存到一个文件夹中；样式表、音频、视频等同样如此。
3. 打开电子书2，在Text文件中选择任意一个内容文档，点击鼠标右键，在菜单中选择「添加现有文件」。
4. 在弹出窗口中可以选择内容文档、图片、样式表等，然后点击「打开」按钮，Sigil会把文件分别放置在合适的文件夹中。
5. 按照章节的顺序，对内容文档进行排序。

添加现有文件，注意内容文档的文件名不要相同，如果是相同的，会提示你替换或取消。如果它们的内容不同，你需要在电子书1中重新命名内容文档，然后再重复步骤1。

附录一

epub:type属性定义

本内容来自《ePub3.0内容文件规范中文版》第2章，感谢翻译人员，在网上搜索书名即可下载！

语义转折是用来处理在XHTML内容文件中的元素，其特定用途或本质之外所额外附加的意义。在ePub出版物的内文中，`epub:type`属性通常用以表达特定内容的语义，透过随之而来的语义转折去补足底层[HTML5]主机字（host vocabulary）。所套用的语义会改变所包含元件的意思，但不会覆盖其本质。

例如，该属性可以用来将一件作品中的section指定为一个chapter（章节），但是不能用来把p元素转变为清单项目，以避免适当的清单结构。

语义书目元数据（semantic metadata）并非是要给人们使用的，而是要让阅读系统及其他使用者能够进一步了解文件的结构和内容，以提升使用者阅读体验。XHTML内容文件中增加XML元件，而是以`epub:type`属性附加在已知的元件上，用以变化为所需的语义。

可见10.6.4节。范例展示了`epub:type`属性用以转折注脚与注释参照语义。所使用的性质皆定义于预设词汇里。

以下范例展示`epub:type`属性用以转折来源出版物分页语义。所使用的性质皆定义于预设词汇里，指定适用该页码信息的来源出版物。

```
<html ... xmlns:epub="http://www.idpf.org/2007/ops">
  ...
  <p> ... <span epub:type="pagebreak" title="234"></span> ... </p>
  ...
</html>
```

属性定义

属性分类	属性定义	属性说明
文档划分	cover	出版物的封面、护封等
frontmatter	题词	
bodymatter	主体	
backmatter	附录	
文档分隔	volume	卷
part	部分	
chapter	章	
subchapter	节	
division	段	
文档组成部分	epigraph	相关引用
conclusion	结论	
afterword	编后记	
warning	警告	
epilogue	后记	
foreword	前言，序	
introduction	介绍，引言	
prologue	序言	
preface	引语	
preamble	绪言	
notice	通知、布告	
参考部分	landmarks	参考文献
lot	贡献者	
index	索引	
colophon	版本记录	
appendix	附录，增补信息	
loi	插图说明	
toc	目录	
词汇表	glossary	词汇表
glossterm	词汇期限	
glossdef	注释	
书目提要	biblioentry	书目提要
bibliography	引用列表	
开始部分	imprint	版权标记
errata	勘误表	
copyright-page	版权页	
acknowledgments	感谢	
other-credits	其它荣誉	
titlepage	扉页，书名页	
imprimatur	出版许可	
contributors	贡献者	
halftitlepage	简名页	
dedication	题词	
补充内容	help	帮助
sidebar	增补内容	
annotation	注释	
marginalia	旁注	
practice	练习或样本	
注释	note	注释
footnote	注脚	
rearnote	后部注释	
footnotes	注脚	
rearnotes	后部注释	
题头	bridgehead	副标题
标题	title	标题
halftitle	首页标题	
fulltitle	标题和子标题的组合	
subtitle	副标题	
covertitle	封面标题	

文档文本	concluding-sentence	段落结语
keyword	关键词	
topic-sentence	引言结语	
参考	annoref	注释参考
noteref	脚注参考	
页码标记	page-list	页面列表
pagebreak	分页	
表单	table	表单
table-row	表单行	
table-cell	单元格	
列表	list	列表
list-item	列表的条目	

可以参考上面的用法添加到文中相应的内容元素中。

附录二

多看阅读的教案

本节内容是从QQ群收集的，非作者原创，也获得了官方的转载许可。该教案与本书有重叠，所以在一些内容中有参考某某某节字样，特此说明。部分内容与原教案不同，采用了更加清晰明了的表格展示方式。

本规范需要使用多看阅读的编辑工具——Dobby。搜索QQ群「Dobby守护精灵群」，申请加入，从群文件中获取安装包。那么，让我们来看下正规公司的标准制作流程以及规范吧！

关于多看阅读器的更多扩展效果，请参考以下章节：交互图（10.4.2节）、画廊模式（10.4.4节）、透明图昼夜自适应（10.4.7节）、封面全屏（10.5.1节）、弹注（10.6.1和10.6.3节）、音频（10.7.1节）、视频（10.7.2节）、竖排（10.9.2节），其他扩展效果（图片贴边、全屏展开、波浪形下划线、伪·正版书效果、调用阅读器字体、漫画格式）另见10.11节，多看阅读器的听书格式（11章）。

图片规范

Dobby支持的图片格式：jpg格式和png格式。

全屏图

全屏图即图片充满屏幕全部的图片。分为封面图、章节页图、其他全屏图：

1. 封面图：电子书第一页用做封面的图片。
2. 章节页图：第一章、第一节或栏目一、栏目二通常也会做成的全屏图。
3. 其他全屏图：全屏插画、广告页等会用到的全屏图片。

尺寸



4:3屏：768px宽×1024px高　　　16:9屏：720px宽×1280px高

命名（XX代表章节或纸版页码的数字）

封面图。统一为：

- 4:3屏：coverpage.jpg
- 16:9屏：coverpage~slim.jpg

章节页图。统一为：

- 4:3屏：chapterXX.jpg
- 16:9屏：chapterXX~slim.jpg

其他全屏图：

- 4:3屏：XX-f.jpg
- 16:9屏：XX-f~slim.jpg

适配3:2屏幕的4:3全屏图

3:2屏幕的全屏图使用的是4:3尺寸的全屏图。但因尺寸不同，3:2的屏幕在显示4:3全屏图时会显示不全。

有文字的全屏页图片，左右要各留出至少100px的宽度，防止在3:2的屏幕上显示的时候，距离屏幕左右边缘过近的文字被隐藏掉。

原理如下图所示，红框范围是4:3屏幕所能显示的范围，黄框内是3:2的屏幕显示的范围：



全屏图中主要的文字和图片不要超出黄框以外。一旦超出的话，超出的内容在3:2的屏幕上会隐藏起来。

3:2的屏幕多见于iPad（横屏双页）、小米平板（横屏双页）、iPhone4系列等。

非全屏图

非全屏图主要指内文页面中分布在文字周围的图片，分为横图（宽度 > 高度）、竖图（宽度 < 高度），宽度≈高度的方图按竖图处理。

尺寸

横图：1152px宽×等比例缩放高度值（1152px×768px为佳）；

竖图：768px宽×等比例缩放高度值（768px×1152px为佳）。

命名

建议以三位数数字命名图片，这里数字代表纸书页码数，方便对照纸书，如：001.jpg、002.jpg，则分别代表第一页和第二页的两张图。

如果同一页有多张图片，则命名时用「-1」、「-2」、「-3」加在后面，表示每页中每张图的顺序，如第一页有三张图，则分别命名为：001-1.jpg、001-2.jpg、001-3.jpg……

交互图命名

页码数后加「-i」，如第10页第2张图是交互图时，则命名为：010-2-i.jpg。

非交互图命名

背景不透明的图，存储成jpg格式，命名为「页码数.jpg」，如第11页第3张图，则命名为：011-3.jpg。

背景透明的图，存储成png格式，命名为「页码数.png」，如第11页第3张图，则命名为：011-3.png。

图片与文件命名时所用符号规则

允许使用的字符

半角英文、半角数字、半角短横线「-」、半角下划线「_」。

特例

16:9全屏图的「～」，除此以外，空格、半全角括号、全角符号、中文都是不被允许使用的。

命名顺序

图片命名必须是字母或数字开头；文件命名必须是字母开头。

大小写

Dobby对文件名大小写非常敏感，非常推荐所有文件使用小写扩展名。

路径

Dobby电子书制作路径：

多看电子书

```
└书名文件夹
    ├A 数据源——PDF、INDD、TXT、WORD
    ├B 编辑加工——当天日期的文件夹，比如：20141014
    └C 图片设计
        ├章节图文件夹
        └内文图文件夹
```

用户自建CSS样式说明

样式，即对文字、图片、页面进行外观修饰，格式美化的命令。下文为用户自建样式参考。

样式类型

h：文字标题，按层级由高到低分别是h1、h2、h3、h4、h5、h6，见2.1.3节；

p：文字段落，见2.1.1节；

span：文字标题或文字段落或单元格内的一部分文字，见2.2.2节；

div：区域块，见2.1.2节；

img：图片，见2.2.10节；

ol：有序的总列表，见2.1.5节；

ul：无序的总列表，见2.1.5节；

li：列表中的每一个单项（Dobby中目前li不需要样式）；

pre：代码块（通常使用软件自带样式即可），见2.1.10节；

table：表格，见2.1.4节；

tr：每一行（Dobby中目前tr不需要样式）

td：每一行中的单元格

body：页面，通常设置背景，见2.3.6节；

a：对部分文字、图片设置超链接，见2.2.9节；

CSS结构

注：本教程中均有此类实例，可参考。

```
p.bodycontent-text {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size: 14px;
    text-align: center;
```

```
}
```

分解说明（「{}」内各种丰富多彩的样式命令请见下一小节）：

表30　样式的组成

p.bodycontent-text {}					
p	.	bodycontent-text		{	}
样式类型	半角下圆点	样式的英文调用名称	半角空格	左花括号	右花括号

各种样式类型内部的样式命令

文字标题h、文字段落p：

`font-family: "DK-KAITI", "方正楷体";`

字体10种，见10.11.5一节；

以下内容见3.4.14节。

`font-size: 16px;`，字号。单位px，像素。默认16px；

`font-weight:  normal;`，字型。可选为normal（正常）或bold（加粗），字型可以不用填，但h类型样式默认为bold加粗；

`font-style:   italic;`，每一个文字都是斜体。需要设置为斜体则填写，不需要则不用写；

`color: #91531d;`，文字颜色。「#91531d」是十六位颜色代码；

`text-align:    left;`，文字水平方向的对齐，四种可选：left（强制左对齐）、center（居中）、right（居右）、justify（两端对齐，软件默认）；

`text-indent`与`duokan-text-indent`：首行缩进。这两个命令在写的时候必须同时写上，并且数值相同，一般为2em或0em。强制居左或两端对齐的时候必须写，居中或居右时则不要写本命令；

border　边框

边框写法1：四边相同边框统一设置

`border: solid 1px #91531d;`，见3.4.3一节。

「solid」为单实线，「1px」是粗细，「#91531d」是十六位颜色代码。线型：solid单实线、double双实线、dashed虚线、dotted点线、none无。

边框写法2：四边不同边框分别设置

`border-top: solid 1px #91531d;`上边框；

`border-right: double 1px #91531d;`右边框；

`border-bottom: double 1px #91531d;`下边框；

`border-left: solid 1px #91531d;`左边框。

边框写法3：四边不同边框分别设置

`border-style: solid double dashed dotted;`，代表上边框是单实线，右边框是双实线，下边框是虚线，左边框是点线。

`border-width: 10px 6px 2px 20px;`，上、右、下、左边框粗细分别为10px、6px、2px、20px。

`border-color: #a0a0a0 #a0a0a0 #a0a0a0 #ff00ff;`，上、右、下、左边框的颜色。

边框写法4：四边边框相同的部分统一设置，不同的部分分别设置

`border-style: solid none solid none;`，代表上下边框是单实线，左右边框是无。

`border-width: 10px 6px 2px 20px;`，上、右、下、左边框粗细分别为10px、6px、2px、20px。

`border-color: #a0a0a0;`，四边边框的颜色都是灰色。

`background-color: #f1d2c3;`，背景色。

`border-radius: 1em 0.3em 0.4em 0em;`，边框圆角弧度设置（0em~1em）。四组数值分别代表左上角、右上角、右下角、左下角，必须搭配`border`或`background-color`使用。

padding 内边距

见3.4.9节。

内边距写法1：四边相同内边距统一设置

`padding 0.5em;`，「0.5em」代表四边的内边距统一为半个字宽度。

内边距写法2：四边内边距分别设置（每个方向在设置时，需要就写，不需要则可以不写）

```
padding-top: .5em;
padding-right: .3em;
```

```
padding-bottom: .4em;
padding-left: .8em;
```

内边距写法3：四边内边距分别设置（每个方向的设置都必须写）

`padding:   0.5em   0.3em   0.4em   0em;`，上、右、下、左内边距分别为0.5em、0.3em、0.4em、0em。

margin 外边距

见3.4.8节。

外边距写法1：四边外边距分别设置（每个方向的设置都必须写）

`margin:   2em   1em   1.5em   0em;`，上、右、下、左外边距分别为2em、1em、1.5em、0em。

外边距写法2：四边外边距分别设置（每个方向的设置，需要就写，不需要则可以不写）

```
margin-top: 2em;
margin-right: 1em;
margin-bottom: 1.5em;
margin-left: 0em;
```

外边距的居中特例：左右外边距同时设置为auto，并搭配width。

```
margin-left: auto;
margin-right: auto;
width: 80%;
```

宽度。数值单位可以为百分比，也可以为em。单位为em时必须搭配font-size设置本样式的基础字号。

`text-decoration:            underline;`，每一个文字的装饰线。可选为underline（下划线）、overline（上划线）、line-through（单删除线）。

`text-shadow: 0.05em 0.06em 0.02em #808080;`，每一个文字的阴影。四组数值分别代表向右位移0.05个字宽度、向下位移0.06个字宽度、阴影离散度（0em～1em）、阴影颜色。

文字标题h和文字段落p的样式命令中必须写的有：`font-family`和`font-size`。需要注意的是`text-indent`与`duokan-text-indent`需要同`text-align`搭配使用。

span

表31　span元素支持的属性（多看）

支持属性		不支持属性	
属性	含义	属性	含义
font-family	字体	text-align	文字水平对齐方向
font-size	字号	text-indent	公共CSS规范中的首行缩进
font-weight	文字正常或加粗	duokan-text-indent	多看CSS规范中的首行缩进
font-style	文字是否斜体	padding	内边距
color	文字颜色	margin	外边距
background-color	文字本身的背景色	border	边框
text-decoration	文字装饰线	border-radius	边框圆角
text-shadow	文字阴影	（不支持的就不要写啦）	

> 还有用处比较大的行内（span）背景，也属于不支持范畴。

div

支持写上去的有：

表32　div元素支持的属性（多看）

属性	含义	属性	含义
font-size	给单位是em的命令设置基础字号	text-align	给本区域内部元素（h、p、img）设置水平对齐方向
border	边框	padding	内边距
background-color	本区域的背景色	border-radius	边框圆角
margin	外边距	width	宽度

img

支持写上去的有：

表33　img元素支持的属性（多看）

属性	含义	属性	含义
font-size	给单位是em的命令设置基础字号	width	宽度。单位为百分比或em字宽。高度和宽度只能写一个，不能同时写。
margin	外边距	height	高度。单位为em字宽。高度和宽度只能写一个，不能同时写。
vertical-align	垂直方向对齐设置可选为top（顶部对齐）、duokan-middle-line（居中对齐）、bottom（底部对齐）。也可用em为单位填写具体对齐数值。		

ol

支持写上去的有：

表34　ol元素支持的属性（多看）

属性	含义	属性	含义
font-size	文字字号	font-family	字体
margin	外边距		

ul

支持写上去的有：

表35　ul元素支持的属性（多看）

属性	含义	属性	含义
font-size	文字字号	font-family	字体。目前支持list-style-type种类最多的字体是楷体
margin	外边距	list-style-type	项目符号类型。可选为none（无）、disc（实心圆点）、circle（空心圆圈）、square（实心方块）

table

见2.1.4和3.4.13二节。

支持写上去的有：

表36　table元素支持的属性（多看）

属性	含义	属性	含义
font-size	给单位是em的命令设置基础字号	background-color	全表背景色，包括单元格间隙
有框表格需要用这种写法设置边框，border-style的边框有无顺序为固定写法（无框表格则可以不写） border-style: none solid solid none; border-width: 1px; border-color: #000000;			

td

支持写上去的有：

表37　td元素支持的属性（多看）

属性	含义	属性	含义
font-size	文字字号	font-family	字体
text-align	文字水平对齐方向	duokan-text-indent	多看CSS规范中的首行缩进
text-indent	公共CSS规范中的首行缩进	background-color	单元格内部的背景色，不含单元格间隙
padding	内边距	border	边框
vertical-align	垂直对齐方向。td中的vertical-align可		

	选为top、middle、bottom，与img中的略有差异
有框表格需要用这种写法设置边框，border-style的边框有无顺序为固定写法（无框表格则可以不写）	
border-style: none solid solid none;	
border-width: 1px;	
border-color: #000000;	

width单写在xhtml页面`<td>`中的写法有两种，请注意写法差异：

第一种，百分比为单位：`<td width="20%"></td>`；

第二种，em为单位：`<td style="width: 12em"></td>`。

rowspan：某个单元格与其所在列的下一行或几行合并，在td内写法为`<td rowspan="3"><br /></td>`；

colspan：某个单元格与其所在行的右边一列或几列合并，在td内的写法为`<td colspan="3"><br /></td>`。

body

body一般不用单独设置CSS，只有需要设置body的时候才写进CSS中，目前支持写上去的有：

背景纯色：`background-color: #f1d2c3;`

背景图（以下命令同时写）：

```
background-image: url("../Images/bg.jpg"); /*「bg.jpg」是已添加的背景图*/
text-align: justify;
background-size: cover;
background-position: center;
```

a

支持写上去的有：

表38　a元素支持的属性（多看）

属性	含义	属性	含义
color	链接颜色，包含下划线装饰	font-style	是否斜体
text-decoration	设置装饰线		

字体

见10.11.5节。

元数据编辑器相关说明

1. 标题：填写标准书名；
2. 作者与编者：「著」字样，归为作者；「编」、「编著」、「主编」字样，归为编者；
3. 插图作者：这个我们一般很少填写，一般就略过就行；
4. 译者：填写译者，有就写，没有就不写（注：古文今译的「译」划归为作者）；
5. 出版商：请使用标准名称，如：华章在我们数据库里用的是北京华章图文信息有限公司；
6. 封面图：请使用那张coverpage.jpg的图片作为元数据封面。

关于人名：

1. 外国籍需要在每个人名前面加上用【】括起的国籍简称；如果中英文名字都有，需中文在前，英文名字在后，英文名字用全角括号（）括起；如果有英文名字缩写，需小数点后面加一个半角空格与后面的英文字母隔开，如：【德】卡尔·马克思（Karl H. Marx）
2. 多个人名之间用一个半角分号和一个半角空格隔开，如：【德】卡尔·马克思（Karl H. Marx）; 【德】弗里德里希·恩格斯（Friedrich V. Engels）
3. 中国朝代用全角括号括起，放在人名前面，如：（明）徐光启
4. 关于「等」字，填写时应与人名紧贴，如：王小波; 李银河等
5. 两个字的名字，中间不要有空格，如：莫言

这类元数据也可以自己拟定一个标准，后续制作的书只要遵循这个标准即可。

默认样式表

本样式表取自Dobby软件，如有侵权，请与我联系删除。

```
/* 西文字体 */
span.duokan-western {
    font-family: "DK-SERIF", "Palatino";
}
/* 符号字体 */
span.duokan-symbol {
    font-family: "DK-SYMBOL", "Symbol";
}
/* 文字不换行（古诗） */
span.duokan-text-nowrap {
    white-space: nowrap;
}
/* 段首大字2行高 */
span.duokan-dropcaps-two {
    float: left;
    font-size: 2em;
    duokan-drop-caps-style: duokan-two-line-caps;
```

```
    margin-bottom -0.25em;
}
/* 段首大字3行高 */
span.duokan-dropcaps-three {
    float: left;
    font-size 3em;
    duokan-drop-caps-style: duokan-three-line-caps;
    margin-bottom -0.15em;
}
/* 代码中文样式 */
span.duokan-code-cn {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
}
/* 超链接 */
span.duokan-hyperlink {
    color: blue;
    font-style: italic;
    text-decoration: underline;
}
/* 多看注 */
span.duokan-footnote {
    width .8em;
}
/* 默认有序列表 */
span.duokan-olist {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
}
/* 无序号有序列表 */
span.duokan-olist-noindex {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    list-style-type: none;
}
/* 默认无序列表 */
span.duokan-ulist {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    list-style-type: disc;
}
/* 二级列表 */
span.list-sub2 {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    list-style-type: circle;
    margin-left 1em;
    margin-top 1em;
```

```
}
/* 三级列表 */
span.list-sub3 {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    list-style-type: square;
    margin-left 2em;
    margin-top 1em;
}
/* 图片左绕排50%宽度 */
span.duokan-float-left {
    width 50%;
    float: left;
    margin-right .5em;
    margin-bottom .5em;
    text-align: center;
}
/* 图片左绕排30%宽度-细高图 */
span.duokan-float-left30 {
    width 30%;
    float: left;
    margin-right .5em;
    margin-bottom .5em;
    text-align: center;
}
/* 图片右绕排50%宽度 */
span.duokan-float-right {
    width 50%;
    float: right;
    margin-left .5em;
    margin-bottom .5em;
    text-align: center;
}
/* 图片右绕排30%宽度-细高图 */
span.duokan-float-right30 {
    width 30%;
    float: right;
    margin-left .5em;
    margin-bottom .5em;
    text-align: center;
}
/* 图片居中80%宽度 */
span.duokan-center {
    width 80%;
    margin: 1em auto 1em auto;
    text-align: center;
}
```

```
/* 图片居中60%宽度 */
span.duokan-center60 {
    width 60%;
    margin: 1em auto 1em auto;
    text-align: center;
}
/* 图片居中30%宽度 */
span.duokan-center30 {
    width 30%;
    margin: 1em auto 1em auto;
    text-align: center;
}
/* 图片-无图说 */
span.duokan-image {
    width 100%;
}
/* 图片-有图说 */
span.duokan-image-note {
    width 100%;
    font-size 16px;
    margin-bottom .5em;
}
/* 图说楷体居左顶格 */
span.duokan-note {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 14px;
    text-indent 0em;
    duokan-text-indent 0em;
    text-align: justify;
}
/* 图说楷体居中 */
span.duokan-note-center {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 14px;
    text-align: center;
}
/* 图说楷体居右 */
span.duokan-note-right {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 14px;
    text-align: right;
}
/* 表格 */
span.duokan-tablebody {
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-weight: normal;
    font-size 14px;
```

```
    margin-top 1em;
    margin-bottom 2em;
    margin-left: auto;
    margin-right: auto;
}
/* 并列图表格样式 */
span.duokan-gallery-image {
    font-size 1em;
    width 100%;
    margin: 1em auto 1em auto;
    border-spacing: .3em .3em;
}
/* 并列图单元格样式 */
span.duokan-gallery-cell {
    text-indent 0em;
    duokan-text-indent 0em;
}
/* 格内图样式 */
span.duokan-in-cell {
    width 100%;
}
/* 代码样式 */
pre.duokan-code {
    font-family: "DK-CODE", "Inconsolata";
    font-size 14px;
}
/* 文段内-部分代码字体 */
span.duokan-nowestern-code-span {
    font-family: "DK-CODE", "Inconsolata";
    color: #91531d;
}
/* 文段内-部分代码字体-加粗 */
span.duokan-nowestern-code-span-b {
    font-family: "DK-CODE", "Inconsolata";
    color: #91531d;
    font-weight: bold;
}
/* 文段内-部分代码字体-斜体 */
span.duokan-nowestern-code-span-i {
    font-family: "DK-CODE", "Inconsolata";
    color: #91531d;
    font-style: italic;
}
/* 文段内-部分代码字体-粗体斜体 */
span.duokan-nowestern-code-span-b-i {
    font-family: "DK-CODE", "Inconsolata";
    color: #91531d;
```

```
    font-weight: bold;
    font-style: italic;
}
/* 代码块-部分加粗 */
span.code-b {
    font-weight: bold;
    color: #91531d;
}
/* 代码块-部分斜体加粗 */
span.code-italic-b {
    font-style: italic;
    font-weight: bold;
    color: #91531d;
}
/* 代码块-部分注释色 */
span.note {
    color: #008000;
}
/* 代码块-部分注释色加粗 */
span.note-b {
    color: #008000;
    font-weight: bold;
}
/* 版权信息页标题 */
span.duokan-copyright-title {
    font-size 24px;
    font-family: "DK-XIAOBIAOSONG", "方正小标宋简体";
    font-weight: normal;
    text-align: center;
}
/* 版权信息页正文 */
span.duokan-copyright-bodytext {
    font-family: "DK-SONGTI", "方正宋三简体", "方正书宋", "宋体";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 拼音样式 */
ruby {
    ruby-align: center;
    margin-right .5em;
}
/* 拼音尾字样式 */
ruby.duokan-rubylast {
    margin-right 0em;
}
/* 拼音文字样式 */
```

```css
ruby > rt {
    font-family: "DK-SYMBOL", "Symbol";
    font-size .5em;
}
/* 一级标题小标宋居中 */
span.auxiliary-title1 {
    font-family: "DK-XIAOBIAOSONG", "方正小标宋简体";
    font-size 28px;
    text-align: center;
    color: #91531d;
    font-weight: normal;
    margin-top 2.5em;
    margin-bottom 2.5em;
}
/* 二级标题 */
span.chapter-title2 {
    font-family: "DK-XIAOBIAOSONG", "方正小标宋简体";
    font-size 26px;
    text-indent 0em;
    duokan-text-indent 0em;
    color: #91531d;
    font-weight: normal;
    margin-top 2em;
    margin-bottom 1.8em;
}
/* 三级标题 */
span.chapter-title3 {
    font-family: "DK-XIAOBIAOSONG", "方正小标宋简体";
    font-size 22px;
    text-indent 0em;
    duokan-text-indent 0em;
    color: #91531d;
    font-weight: normal;
    margin-top 2em;
    margin-bottom 1.8em;
}
/* 内文四级标题 */
span.chapter-title4 {
    font-size 20px;
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-weight: normal;
    duokan-text-indent 0em;
    text-indent 0em;
    margin-top 2em;
    margin-bottom 1.8em;
    color: #91531d;
}
```

```
/* 内文五级标题 */
span.chapter-title5 {
    font-size 18px;
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-weight: normal;
    duokan-text-indent 0em;
    text-indent 0em;
    border-left: solid 5px #d9d9d9;
    padding-left .7em;
    margin-top 2em;
    margin-bottom 1.8em;
    color: #91531d;
}
/* 内文六级标题 */
h6.chapter-title6 {
    font-size 16px;
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-weight: normal;
    duokan-text-indent 0em;
    text-indent 0em;
    margin-top 2em;
    margin-bottom 1.8em;
    margin-left 2em;
    color: #91531d;
}
/* 正文宋体 */
span.bodycontent-text {
    font-family: "DK-SONGTI", "方正宋三简体", "方正书宋", "宋体";
    font-size 16px;
    duokan-text-indent 2em;
    text-indent 2em;
}
/* 正文宋体-强制左对齐 */
span.bodycontent-text-left {
    font-family: "DK-SONGTI", "方正宋三简体", "方正书宋", "宋体";
    font-size 16px;
    text-align: left;
    duokan-text-indent 2em;
    text-indent 2em;
}
/* 正文宋体-强制左对齐顶格 */
span.bodycontent-text-left0 {
    font-family: "DK-SONGTI", "方正宋三简体", "方正书宋", "宋体";
    font-size 16px;
    text-align: left;
```

```
    duokan-text-indent 0em;
    text-indent 0em;
}
/* 引文黑体标题居中 */
span.reference-title {
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-size 16px;
    text-align: center;
}
/* 引文楷体 */
span.reference-text {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 引文楷体顶格 */
span.letter-head {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    duokan-text-indent 0em;
    text-indent 0em;
}
/* 署名楷体居右 */
span.signature {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    text-align: right;
}
/* 诗集编号小标宋居中 */
span.num-tilte {
    font-family: "DK-XIAOBIAOSONG", "方正小标宋简体";
    font-weight: normal;
    font-size 18px;
    margin-bottom 5em;
    color: #91531d;
    text-align: center;
    margin-top 2em;
}
/* 部分斜体 */
span.xieti {
    font-style: italic;
}
/* 部分斜体加粗 */
span.italic-b {
    font-style: italic;
    font-weight: bold;
```

```
}
/* 部分加粗 */
span.bold {
    font-weight: bold;
}
/* 部分楷体 */
span.kaiti {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
}
/* 部分楷体加粗 */
span.kaiti-jiacu {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-weight: bold;
}
/* 部分黑体加粗 */
span.heiti {
    font-family: "DK-DENGXIAN";
    font-weight: bold;
}
/* 楷体自然段 */
span.kaiti-duan {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 楷体加粗自然段 */
span.kaiti-duan-jiacu {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
    font-weight: bold;
}
/* 黑体加粗自然段 */
span.heiti-duan {
    font-family: "DK-DENGXIAN";
    font-weight: bold;
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 下划线 */
span.xiahuaxian {
    text-decoration: underline;
}
/* 音频 */
```

```
audio.yinpin {
    height 1.2em;
    vertical-align: duokan-middle-line;
}
/* 视频 */
video.shipin {
    width 100%;
    margin: 1em auto 1em auto;
    text-align: center;
}
/* 1字高图标 */
span.formula-1em {
    height 1em;
    vertical-align: duokan-middle-line;
}
/* 1.5字高图标 */
span.formula-1-5em {
    height 1.5em;
    vertical-align: duokan-middle-line;
}
/* 2字高图标 */
span.formula-2em {
    height 2em;
    vertical-align: duokan-middle-line;
}
/* 2.5字高图标 */
span.formula-2-5em {
    height 2.5em;
    vertical-align: duokan-middle-line;
}
/* 填充的点着重号 */
span.emphasis-filled-dot {
    text-emphasis: filled dot;
}
/* 上标字体 */
span.math-super {
    font-size .7em;
    vertical-align: super;
}
/* 下标字体 */
span.math-sub {
    font-size .7em;
    vertical-align: sub;
}
/* 上标倾斜字体 */
span.math-super-italic {
    font-size .7em;
```

```
    font-style: italic;
    vertical-align: super;
}
/* 下标倾斜字体 */
span.math-sub-italic {
    font-size .7em;
    font-style: italic;
    vertical-align: sub;
}
/* 无序号无序列表 */
span.duokan-ulist-noindex {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    list-style-type: none;
}
/* 黑体强调 */
span.font-heiti {
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    color: #91531d;
}
/* 楷体强调 */
span.font-kaiti {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    color: #91531d;
}
/* 斜体 */
span.italic {
    font-style: italic;
}
/* 章前语带框线背景 */
span.chapter-pre-div0 {
    font-size 16px;
    padding: .5em .5em .5em 1em;
    margin: 15% 0em 1em 0em;
    border-left: solid 1.5px #e0e0e0;
}
/* 章前语无装饰 */
span.chapter-pre-div {
    font-size 16px;
    padding: .5em .5em .5em .5em;
    margin: 15% 0em 1em 0em;
}
/* 章前语灰色底色 */
span.chapter-pre-div2 {
    font-size 16px;
    padding .5em;
    background-color: #f4f4f4;
```

```css
    margin-bottom 1.5em;
}
/* 章前语标题 */
span.chapter-pre-title2 {
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-size 16px;
    text-indent 0em;
    duokan-text-indent 0em;
    margin-bottom .8em;
    margin-left 1em;
}
/* 章前语文本 */
span.chapter-pre-text {
    font-family: "DK-SONGTI", "方正宋三简体", "方正书宋", "宋体";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 代码清单 */
span.code-list {
    font-family: "DK-SONGTI", "方正宋三简体", "方正书宋", "宋体";
    font-size 16px;
    duokan-text-indent 0em;
    text-indent 0em;
    padding: .4em 0em .23em .5em;
    background-color: #e0e0e0;
    border-bottom: solid 1px #c0c0c0;
    margin-top 1.6em;
    margin-bottom .8em;
}
/* 知识点div */
span.knowledge-point-div {
    font-size 16px;
    padding .5em;
    margin: 1.5em 0em 1.5em 0em;
    border-style: solid none solid none;
    border-width 1px;
    border-color: #909090;
}
/* 底纹知识点div */
span.knowledge-point-div-bgc {
    font-size 16px;
    padding .5em;
    margin: 1.5em 0em 1.5em 0em;
    background-color: #f4f4f4;
    border-bottom: solid 1px #c0c0c0;
```

```css
}
/* 底纹知识点标题 */
span.knowledge-point-title-bgc {
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    padding: .4em 0em .23em .3em;
    margin-top -0.5em;
    margin-left -0.5em;
    margin-right -0.5em;
    margin-bottom 1em;
    font-size 16px;
    text-align: center;
    color: #000000;
    border-bottom: solid 1px #c0c0c0;
    background-color: #e0e0e0;
}
/* 知识点标题 */
span.knowledge-point-title {
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-size 16px;
    duokan-text-indent 0em;
    text-indent 0em;
    color: #91531d;
}
/* 知识点二级标题 */
span.knowledge-point-title2 {
    font-family: "DK-HEITI", "方正兰亭黑简体", "黑体";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
    margin-top 2em;
    color: #91531d;
}
/* 知识点文段 */
span.knowledge-point-text {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 知识点引文仿宋 */
span.knowledge-point-reference {
    font-family: "DK-FANGSONG", "方正仿宋", "华文仿宋", "仿宋";
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 楷体加粗强调 */
```

```
span.font-kaiti-b {
    font-family: "DK-KAITI", "方正楷体", "华文楷体", "楷体";
    color: #91531d;
    font-weight: bold;
}
/* 下单实线文段 */
span.border-botttom-solid {
    font-family: "DK-SONGTI", "方正宋三简体", "方正书宋", "宋体";
    border-bottom: solid 1px #000000;
    margin-top 0em;
    font-size 16px;
    text-indent 2em;
    duokan-text-indent 2em;
}
/* 上下点线装饰区域 */
span.border-top-bottom-dotted {
    border-top: dotted 1px #000000;
    border-bottom: dotted 1px #000000;
    padding .2em;
}
```

附录三

自用字体样式表

样式作者：阜東子安

样式版本：0.81

样式日期：2016.1.30（首次使用）2017.4.7（最后修改）

参考资料：http://www.w3school.com.cn/css3/css3_font.asp

样式说明：本样式在「阜東子安」字体样式的基础上修改，仅用于赤霓自制书。

```
/*方正屏显书宋_GBK 版本1.00*/
@font-face {
  font-family: "宋体";
  src: local("st"),local("ST"),local("SongTi"),local("Songti"),
       local("正文"),local("宋体"),local("明体"),local("明朝"),
       local("Songti"),local("Songti SC"),local("Songti TC"),
       local("STSongti"),local("STSong"),local("Song S"),
       local("STBShusong"),local("TBMincho"),
       local("HYMyeongJo"),local("DK-SONGTI"),
  url("../Fonts/SongTi.ttf");
}
/*方正仿宋_GBK 版本5.30*/
@font-face {
  font-family: "仿宋";
  src: local("fs"),local("FangSong"),local("Fangsong"),
       local("仿宋"),local("仿宋_GB18030"),local("仿宋_GB2312"),
       local("方正仿宋_GBK"),local("DK-FANGSONG"),
  url("../Fonts/FangSong.ttf");
}
/*方正准圆_GBK 版本1.10*/
@font-face {
```

```
  font-family: "准圆";
  src: local("yt"),local("STYuanti"),local("STYuan"),
       local("Yuanti TC"),local("Yuanti"),local("Yuanti SC"),local("DK-XIHEITI"),
       url("../Fonts/ZhunYuan.ttf");
}
/*方正楷体_GBK 版本5.30*/
@font-face {
  font-family: "楷体";
  src: local("kt"),local("KaiTi"),local("STKaiti"),local("STKai"),
       local("楷体"),local("楷体_GB18030"),local("楷体_GB2312"),
       local("Kaiti"),local("Kaiti SC"),local("Kaiti TC"),
       local("MKai PRC"),local("MKaiGB18030C-Medium"),local("方正楷体_GBK"),
       local("MKaiGB18030C-Bold"),local("DK-KAITI"),
  url("../Fonts/KaiTi.ttf");
}
/*方正黑体_GBK 版本5.30*/
@font-face {
  font-family: "黑体";
  src: local("ht"),local("信黑 W4"),local("信黑"),
       local("微软雅黑"),local("雅黑"),
       local("STHeiti"),local("STHei"),
       local("MYing Hei S"),local("TBGothic"),
       local("Heiti"),local("Heiti SC"),local("Heiti TC"),
       local("DK-HEITI"),
  url("../Fonts/HeiTi.ttf");
}
/*方正小标宋_GBK 版本1.10*/
@font-face {
  font-family: "小标宋";
  src: local("h3"),local("h3 XiaoBiaoSong"),local("Heading3"),local("Title3"),local("标题3"),
       local("方正小标宋_GBK"),local("方正小标宋简体"),local("方正小标宋繁体"),
       local("Songti"),local("Songti SC"),local("Songti TC"),
       local("STSongti"),local("STSong"),local("Song S"),local("Song T"),
       local("STBShusong"),local("TBMincho"),local("HYMyeongJo"),
       local("DK-XIAOBIAOSONG"),
  url("../Fonts/XiaoBiaoSong.ttf");
}
```

附录四

掌阅的排版规范

本节所列样式表取自掌阅的ePub编辑工具——ePubGo，如有侵权，可与我联系删除。关于掌阅排版的学习可以到它的官方制书平台上查看相关教程，上面提到的ePubGo也可以在网站上下载到。

质检标准

文件格式检查

文件格式、命名等是否正确

1、封面页、版权信息页必须严格命名为cover和copyright

2、文件命名尽量简短，序号按照层级、顺序排序，不能颠倒、混乱、冗长

3、图片文件，格式只能为*.png*和*.jpg*

4、内文图片大小不超过1M

5、内文图文件命名中不能出现空格、乱码、特殊符号、中文等

6、内文图为全屏图的，要同时制作phone版图，尺寸与封面图一致。非全屏的内文图则不需要

封面

包括平板封面cover、手机封面cover-phone、书架缩略图coverpage，版权背景图copyright，logo图标、光盘标识

封面图是否正确

1、标准尺寸：cover　1536px×2048px、coverpage　300px×400px、cover-phone 720px×1280px

2、图片无拉伸、变形，四边无裁字情况

封面元素是否清晰正确

1、书名、作者等关键信息清晰可见

2、封面信息与数据源或平台提供的信息保持一致

封面logo是否正确

1、国内版和海外版logo图片使用正确，Logo命名为：`zy-cover-logo.png`

2、logo图与封面的色差明显

3、logo的位置正确选在左下、右下、下方之一

光盘标识有无对应链接

封面图带有光盘、视频版标识的，要在后续的内文检查中核查光盘链接信息

版权信息页

确认版权信息页的信息与版权方提供的内容严格保持一致

书名是否正确

1、与版权方提供的书名、封面显示的书名一致（如平台书名与封面不一致请及时反馈）

2、数据源提供有图书在版编目（CIP）数据的，优先以此为准。

3、有丛书名、分册序号、版本号等的，书写原则为：

　　系列名：书名（分册/分卷/版本）

作者名格式是否正确

1、与版权方提供的作/译/编者等名称一致

2、著对应作者；编、编著、主编副主编对应编者；译对应译者；译注、校注保持不变；绘图、插图、插画等对应绘图

3、作者国籍简称正确，且用【】括住；朝代简称正确，且用（）括住

4、外籍作者同时存在中文译名与外文名的，只保留中文名

5、多个作者间用半角空格隔开，省略的作者用「半角空格+等」表示

其他信息是否正确

1、ISBN或ISSN与数据源文件保持一致（如无可略），ISBN号由10/13

位数字组成；ISSN由8位数字构成

2、版权页各元素的样式需要锁死

版权方/授权是否正确

1、授权方名称，与任务大厅中版权方全称一致

2、授权声明格式：「本书由×××××授权掌阅科技电子版制作与发行」

3、掌阅版权的书，格式为：「本书由掌阅科技电子版制作与发行」

目录

目录层级排序是否正确

1、目录的层级与数据源保持一致

2、目录最多三个层级

3、目录的排序按照版权信息、序言、章节目录、参考文献、后记（跋）正确排序

目录标题文字是否正确

1、文字、标点与数据源保持一致

2、生僻字用□代替，避免装饰性符号、竖排标点等

3、标题有双语的只保留中文

4、双语图书，目录按照数据源录入两版目录：1）中英文内容前后完整分隔开的，目录顺序中文在前，英文在后；2）中英文内容间隔排序的，目录顺序也间隔排列

5、杂志目录按照实际排版内容顺序排序

目录跳转是否正确

1、点击目录能正常跳转到对应章节页面（避免跳转错误和目录遗漏章节）

2、多个目录项不能指向同一页面文件：内容较多的，子目录项需拆分页面；内容较少的，在文内添加id锚点

篇章页

篇章标题是否正确

1、章节序号、标题与数据源、排版目录保持一致

2、标题做字号锁定处理

题记文字是否正确

1、题记文字做字号锁定处理

2、题记内容展示不超过4行（phone版显示），过长部分用弹出注展示，超长的题记文字单独拆分成页

篇章页竖排是否正确

篇章页竖排时：文字自右向左排版、使用竖版标点、数字和英文顺时针旋转90度

内文

前言/序言内容是否正确

1、推荐序、译者序、再版序、导言、作者序、前言/序言等内容标题是否上目录

2、前言的标题是否使用了`h1`标签样式；前言的内容是否使用了仿宋样式

3、前言中的书信、引文、诗歌等特殊内容，是否正确使用了对应样式

标题的内容、样式是否正确

1、各标题按照层级对应使用`h1`、`h2`、`h3`、`h6`标签：篇章标题用`h1`，节标题用`h2`，小节标题用`h3`，内文标题用`h6`或特定的`p`标签（尽量避免使用默认正文的`p`标签＋style样式）；同级别的标题样式要保持一致

2、各标题必须限定行高值，`line-height:1.2em;`

3、各级标题的用字遵循「由大到小、由重到轻、变化有序、区别有秩」，字号`font-size`逐级减小，不能出现下级标题的字号大过上级标题；字体尽量使用系统默认的标题字体，与正文字体区别开

4、各级标题的上边距数值必须明显大于下边距。且各级标题的上下间距值成正比，即`h1`的上下间距值＞`h2`＞`h3`＞`h6`，具体数值可参照《排版规范》

5、非篇章页的标题禁止单独占一个页面，标题后面必须有内容展示

6、页面中标题居首的，检查其是否与页面顶部保持留白（默认值20-30%）

7、标题不能出现漏排、顺序错乱、重复、多排等情况。标题文字要与数据源保持严格一致

8、副标题应以`<br />`换行

9、副标题中带有破折号的，破折号跟随副标题文字换行

10、副标题单独应用小于标题字号的`span`样式（字号缩小、字体一般不变、颜色不变）

正文的内容、样式是否正确

1、正文段落样式要带有`zy-text-type: main-body;`代码，且只能正文段落样式独有

2、篇章页后第一段文字，需要与页面顶部保持留白，默认30%（根据具体情况调节）

3、除一些科技类书籍或特殊设计要求（正文段落可顶格）外，其他书籍的正文段落一律缩进2字符，即`text-indent: 2em;`

4、需要上、下标的序号或数字等，使用`sub`、`sup`样式

5、标题、图说、注释等非正文内容不能与正文内容混在一起

6、注意检查内文是否存在乱码、特殊字符、生僻字等情况

7、纸书的页码数字，页眉页脚文字、页码数字等，审核时注意检查是否混入了正文

8、英文书籍的默认正文段落必须是西文字体

正文的标点是否正确

1、中文书籍，正文段落所有标点应为中文状态下的全角标点，不能出现半角标点（特殊情况除外）

2、正文段落句末，一般应以句号。问号、叹号、省略号等结尾，避免逗号，半角标点等不正确的标点结尾

3、现代诗歌、火星文等特殊文体除外；数据源特殊版式除外；科技学术类书籍中数字每三位数间隔用半角逗号除外

4、英文书籍中，段落标点使用英文的半角标点+半角空格（段末不加空格）

5、容易错误的标点：波浪线，应为全角的～；破折号，应为——

生僻字是否正常显示

1、古文类书籍，审核时需对照数据源检查排版是否遗漏生僻字、造字图

2、生僻字可以用繁体字代替的，可以不做生僻字图

3、生僻字图的格式应为背景透明的`.png`文件，尺寸以能清晰显示、与正文字型相近即可

内文特殊语句是否显示正确

1、首字下沉的段落不需要缩进2字符

2、首段的内容显示不足两行的，不使用首字下沉

3、段落的首字为英文、数字、标点的，不使用首字下沉

4、数据源内容中，部分语句、词语有加粗、变颜色、下划线效果的，检查排版是否添加对应`span`效果

5、采访、问答、戏剧等特殊体裁内容，问答、对话的人名称文字需做加粗或改换更粗字体，人名与对话内容之间要有一个全角空格的距离

6、文字带有拼音的，要检查其字母拼写、声调是否正确

7、正文段落中的部分短句引文，使用`span`楷体样式，并检查引文的开头和结尾是否正确包含在内

8、大段引文，使用单独的段落-楷体样式，其首末段与上下文要设置上下间距值1.3em

书信/日记的样式是否正确

1、书信一般会应用3个样式：书信开头、段落-楷体-无边距、落款右。审核时需注意各书信样式是否漏用

2、祝颂语如「此致敬礼」中，「敬礼」样式要顶格

3、书信落款如果同时存在人名、时间的，用`<br />`将其分隔开

4、日记格式中，日期文字应使用区别于正文的单独样式，并加大上间距值（可顶格）

5、书信/日记中如果有引文、诗歌等引用文字，其字体要与书信日记字体区别开；首末段的间距要设置为1.3em

6、全书内容为书信/日记形式的，各书信样式的字体使用默认正文字体，同时添加`zy-text-type: main-body;`属性

诗歌样式是否正确

1、诗歌标题使用「H6诗歌小标题」样式

2、古诗　七言以下，整句使用「诗歌-居中」或「诗歌-居左」样式；七言以上，将整句拆分成单句，使用「诗歌-居中」或「诗歌-居左」样式

3、词、曲　使用 「段落-楷体-无边距」样式，区分上下阙即可

4、现代诗　使用「诗歌-居左」样式，特殊情况另说

5、所有诗歌、词曲段落，与其他段落保持1.3em的上下间距值

6、诗歌中有分篇的，默认分篇间的间距为1em

7、全书诗歌样式统一

插图样式是否正确

1、横幅图，居中独占，width值视长宽比而定（一般为80%-100%）

2、竖幅图，用width值为80%以下的居中独占样式，或视长宽比用width＜50%的左、右绕排样式

3、图片（无图说）要与上下文保持上下1.3em的间距

4、图片＋图说视为一体的模块，用`div`包起来，必须显示在同一页面，不能分页（特殊情况：图说较长且无法用弹出注展示的；有资料来源类图说的）

5、并列图，图片间要有一定数值的间隙；各图的图说文字分隔清晰；并列图使用一个图说的，图说与上图和下文的间距要符合规范要求

6、行内图、签名图、png图等尺寸较小以及像素不够的图片，要用非交互属性

7、图注和标题，字号默认为14px，标题默认字体为中黑，图注默认字体为楷体

8、插图上有文字的，需要将文字识别出来，以图说形式呈现（全屏页图除外）

表格样式是否正确

1、表格，一般以图片的形式展现，应用样式参考插图；注释说明文字样式，参考图说样式

2、表格的标题位于表格的上部，粗体，居中

3、表格跨页显示时，从第二页开始，子表格右上部应标注有「续表」字样，该文字要用楷体、居右样式

4、表标题＋表格＋表注为一体的模块，其模块内容应在一页显示，避免分页

内文注释样式是否正确

1、数据源的脚注、尾注、行内注等，（无特殊情况）统一排版为弹出注，图标分中英两版，图标前后不能有空格

2、诗词校注类图书，按照数据源版式排版，不做弹出注

3、带有生僻字图、上标下标等多媒体属性的注释，允许使用链接跳转注

4、特殊情况：书籍注释内容较多、分段、带有引文等元素的，可以改为内文上标注释序号，注释内容列于节末尾的跳转链接形式

参考文献样式是否正确

1、一般单独分页排版，使用前言后记（仿宋）样式

2、索引内容不排（不涉及页码的人名索引、名词索引可例外）

附录/后记样式是否正确"

1、附录、跋、后记，标题使用h1标签样式，内容使用前言后记（仿宋）样式

2、多个附录应分页排版，不能混在一起

下载链接是否正确

1、光盘或视频的下载链接应单页排版，不能与正文混在一起，以「光盘下载」上目录

2、检查链接的正确性：链接内容为网盘的，要包括链接＋密码，并登录测试是否可下载；内容为掌阅网盘的，要点击测试文件是否与该书内容相符

内容审查

检查图书内容是否存在涉及政治、民族、宗教等方面的敏感话题内容

是否存在争议话题

1、文革话题。深层次地谈论其中内幕、根源等，乃至延伸到政治范畴全盘否定的

2、支持台独、港独、疆独等鼓吹独立的

3、对当前政权形式持否定态度，乃至仇视、丑化等内容

4、引进版权、港台版以及纯电子版发型的书籍，需加倍谨慎此类内容审核

5、内容中有地图的，或者地图类书籍，凡是显示我国地域，不能缺漏部分地区如台湾、钓鱼岛、南海诸岛等

代码审核

对于排版代码，审核时要参照《排版规范》，检查其是否符合要求

各例样式是否正确

1、样式代码书写，要使用英文字符或英文字符加数字的形式，不能只使用数字

2、英文字符要小写

3、字符与字符或字符与数字之间用横杠隔开，不能使用竖杠或者下划线

4、所有的样式名之前必须有元素名（`h1`、`p`、`div`等标签）

5、所有样式全部都要根据样式的作用附上样式注释名称

6、所有的代码标点必须是英文标点

7、样式命名不能重复

8、同一样式里面属性不能重复出现或相互冲突

9、同一个epub文件中有且只能有一个css样式表

最后检查

进行错误信息检查

1、段落未正常结束

2、错误的半角标点

3、漏刷样式或不存在的样式

4、其他错误

未使用的媒体文件

通过错误信息检查，核对是否存在未使用的图片等文件

默认样式表

电子书排版所用的样式表大同小异，掌阅的样式表唯一可取的是字体引用、phone和pad的适配（在属性前加前缀，如`@phone-font-size: 1em;`，针对手机的；`@pad-font-size: 1.2em;`，这个是针对平板的）。

制作适配掌阅的电子书，需要注意的是不要忘记添加*zhangyue-expansion.xml*文件！

```
/* 默认全书背景色 */
body.* {
    background-color: transparent;
}
/* 楷体 */
span.kaiti {
    font-family: "ZY-KAITI", "方正新楷体";
}
/* 仿宋 */
span.fangsong {
    font-family: "ZY-FANGSONG", "方正仿宋";
}
/* 中宋 */
span.zhongsong {
    font-family: "ZY-XIAOBIAOSONG", "方正小标宋";
}
/* 中黑 */
span.zhonghei {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
}
/* 兰亭黑 */
span.lantinghei {
    font-family: "ZY-HEITI", "方正悠黑";
}
/* 小隶书 */
span.xiaolishu {
    font-family: "ZY-LIBIAN", "方正隶变";
}
/* 书宋二简 */
span.shusongerjian {
    font-family: "ZY-ZW-SONGTI", "方正新书宋";
}
/* 西文衬线体lusitana */
span.lusitana {
    font-family: "ZY-SERIF", "Lusitana";
}
/* 西文等线体questrial */
span.questrial {
    font-family: "ZY-SANSSERIF", "Questrial";
}
```

```
/* 西文等线体（粗）quattrocento */
span.quattrocento {
    font-family: "ZY-SANSSERIF-B", "Quattrocento Sans";
}
/* 西文手写体niconne */
span.niconne {
    font-family: "ZY-HANDWRITTEN", "Niconne";
}
/* 小字 */
span.content-word-small {
    font-size: .7em;
    line-height: 1.2em;
    color: #3fadc8;
}
/* 重点 */
span.content-word {
    color: #3fadc8;
}
/* 重点-中黑 */
span.content-word-zhonghei {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    color: #3fadc8;
}
/* 重点-反色 */
span.content-word-light {
    color: #ffffff;
    background-color: #3fadc8;
    padding: .1em .2em;
}
/* H1标题 */
h1.text-title-1 {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 28px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 30%;
    margin-bottom: 2em;
    text-align: justify;
}
/* H1标题-居中 */
h1.text-title-1-c {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 28px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
```

```
    margin-top: 30%;
    margin-bottom: 2em;
    text-align: center;
}
/* H1标题-居右 */
h1.text-title-1-r {
    font-family: "ZY-XIAOBIAOSONG","方正小标宋";
    font-size: 28px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 30%;
    margin-bottom: 2em;
    text-align: right;
}
/* H2标题 */
h2.text-title-2 {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 24px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 20%;
    margin-bottom: 1.3em;
    text-align: justify;
}
/* H2标题-居中 */
h2.text-title-2-c {
    font-family: "ZY-XIAOBIAOSONG","方正小标宋";
    font-size: 24px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 20%;
    margin-bottom: 1.3em;
    text-align: center;
}
/* H3文中小标题 */
h3.text-title-3 {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 20px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.5em;
    margin-bottom: .8em;
    text-align: justify;
```

```
}
/* H3文中小标题-居中 */
h3.text-title-3-c {
    font-family: "ZY-XIAOBIAOSONG","方正小标宋";
    font-size: 20px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.5em;
    margin-bottom: .8em;
    text-align: center;
}
/* H4文中小标题 */
h4.text-title-4 {
    font-family: "ZY-XIAOBIAOSONG","方正小标宋";
    font-size: 18px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.5em;
    margin-bottom: .5em;
    text-align: justify;
}
/* H6诗歌小标题 */
h6.poem-title {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 16px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.3em;
    margin-bottom: .5em;
    text-align: center;
}
/* H6随文小标题-中黑 */
h6.title-zhonghei {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 16px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.3em;
    margin-bottom: .5em;
    text-align: justify;
}
/* H6随文小标题-中宋 */
h6.title-zhongsong {
```

```css
    font-family: "ZY-XIAOBIAOSONG","方正小标宋";
    font-size: 16px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.3em;
    margin-bottom: .5em;
    text-align: justify;
}
/* H6随文小标题-仿宋 */
h6.title-fangsong {
    font-family: "ZY-FANGSONG", "方正仿宋";
    font-size: 16px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.3em;
    margin-bottom: .5em;
    text-align: justify;
}
/* H6随文小标题-兰亭黑 */
h6.title-lantinghei {
    font-family: "ZY-HEITI","方正悠黑";
    font-size: 16px;
    line-height: 1.2em;
    color: #137daf;
    font-weight: normal;
    margin-top: 1.3em;
    margin-bottom: .5em;
    text-align: justify;
}
/* 默认正文段落 */
p.bodytext {
    font-family: "ZY-ZW-HEITI", "方正悠黑";
    zy-text-type: main-body;
    font-size: 16px;
    text-indent: 2em;
    margin-top: 0;
    text-align: justify;
}
/* 序言后记段落 */
p.preface-text {
    font-family: "ZY-FANGSONG", "方正仿宋";
    font-size: 16px;
    text-indent: 2em;
}
/* 段落-中黑 */
```

```
p.bodytext-zh {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 16px;
    text-indent: 2em;
}
/* 段落-兰亭 */
p.bodytext-lt {
    font-family: "ZY-HEITI","方正悠黑";
    font-size: 16px;
    text-indent: 2em;
}
/* 段落-楷体-无边距 */
p.bodytext-kt-np {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    text-indent: 2em;
}
/* 段落-楷体-首段-无边距 */
p.bodytext-kt-first-np {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    text-indent: 2em;
    margin-top: 1.3em;
}
/* 段落-楷体-末段-无边距 */
p.bodytext-kt-last-np {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    text-indent: 2em;
    margin-bottom: 1.3em;
}
/* 段落-楷体-独段-无边距 */
p.bodytext-kt-only-np {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    text-indent: 2em;
    margin-top: 1.3em;
    margin-bottom: 1.3em;
}
/* 诗歌居左 */
p.poem-left {
    padding-left: 2em;
    margin: 0;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: left;
```

```
}
/* 诗歌居左-首行 */
p.poem-left-first {
    padding-left: 2em;
    margin: 0;
    margin-top: 1.3em;
    font-family: "ZY-KAITI", "羿创楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: left;
}
/* 诗歌居左-末行 */
p.poem-left-last {
    padding-left: 2em;
    margin: 0;
    margin-bottom: 1.3em;
    font-family: "ZY-KAITI", "羿创楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: left;
}
/* 诗歌居左-独段 */
p.poem-left-only {
    padding-left: 2em;
    margin-top: 1.3em;
    margin-bottom: 1.3em;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: left;
}
/* 诗歌居左-上间距 */
p.poem-left-margin {
    padding-left: 2em;
    margin: 0;
    margin-top: 1em;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: left;
}
/* 诗歌居中 */
p.poem-center {
    margin: 0;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
```

```
    text-align: center;
}
/* 诗歌居中-首行 */
p.poem-center-first {
    margin: 0;
    margin-top: 1.3em;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: center;
}
/* 诗歌居中-末行 */
p.poem-center-last {
    margin: 0;
    margin-bottom: 1.3em;
    font-family: "ZY-KAITI", "羿创楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: center;
}
/* 诗歌居中-独段 */
p.poem-center-only {
    margin-top: 1.3em;
    margin-bottom: 1.3em;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: center;
}
/* 诗歌居中-上间距 */
p.poem-center-margin {
    margin: 0;
    margin-top: 1em;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: center;
}
/* 落款右 */
p.inscribed-right {
    margin-bottom: 1.3em;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: right;
}
/* 书信开头 */
```

```css
p.inscribed-left {
    margin-top: 1.3em;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    font-weight: normal;
    text-align: justify;
}
/* 默认图片宽100% */
img.zhangyue-img {
    width: 100%;
}
/* 默认图片高100% */
img.zhangyue-img-h100 {
    height: 100%;
}
/* 默认图片高imaker 100% */
img.zhangyue-img-h {
    height: 100%;
}
/* 默认图片宽80% */
img.zhangyue-img-80 {
    width: 80%;
}
/* 默认图片宽50% */
img.zhangyue-img-50 {
    width: 50%;
}
/* 默认图片宽30% */
img.zhangyue-img-30 {
    width: 30%;
}
/* 图片宽4em */
img.zhangyue-img-w4 {
    width: 4em;
}
/* 图片宽1em */
img.zhangyue-img-w1 {
    width: 1em;
}
/* 图片高3em */
img.zhangyue-img-h3 {
    height: 3em;
    vertical-align: middle;
}
/* 图片高1em */
img.zhangyue-img-h1 {
    height: 1em;
```

```
    vertical-align: middle;
}
/* 默认图片横跨 */
img.zhangyue-img-full {
    width: 160%;
}
/* 默认盒子独占居中100% */
div.zhangyue-c {
    width: 100%;
    margin: 1.3em auto;
    text-align: center;
}
/* 默认盒子独占居中80% */
div.zhangyue-c80 {
    width: 80%;
    margin: 1.3em auto;
    text-align: center;
}
/* 默认盒子独占居中60% */
div.zhangyue-c60 {
    width: 60%;
    margin: 1.3em auto;
    text-align: center;
}
/* 默认盒子独占居中40% */
div.zhangyue-c40 {
    width: 40%;
    margin: 1.3em auto;
    text-align: center;
}
/* 盒子左绕排宽50% */
div.zhangyue-float-phone-l {
    width: 50%;
    float: left;
    margin-right: .8em;
    margin-bottom: .5em;
    text-align: center;
}
/* 盒子右绕排宽50% */
div.zhangyue-float-phone-r {
    width: 50%;
    float: right;
    margin-left: .8em;
    margin-bottom: .5em;
    text-align: center;
}
/* 盒子左绕排宽40% */
```

```css
div.zhangyue-float-phone-l40 {
    width: 40%;
    float: left;
    margin-right: .8em;
    margin-bottom: .5em;
    text-align: center;
}
/* 盒子右绕排宽40% */
div.zhangyue-float-phone-r40 {
    width: 40%;
    float: right;
    margin-left: .8em;
    margin-bottom: .5em;
    text-align: center;
}
/* 盒子左绕排宽30% */
div.zhangyue-float-phone-l30 {
    width: 30%;
    float: left;
    margin-right: .8em;
    margin-bottom: .5em;
    text-align: center;
}
/* 盒子右绕排宽30% */
div.zhangyue-float-phone-r30 {
    width: 30%;
    float: right;
    margin-left: .8em;
    margin-bottom: .5em;
    text-align: center;
}
/* 盒子左绕排宽15% */
div.zhangyue-float-phone-l15 {
    width: 15%;
    float: left;
    margin-right: .8em;
    margin-bottom: .5em;
    text-align: center;
}
/* 盒子右绕排宽15% */
div.zhangyue-float-phone-r15 {
    width: 15%;
    float: right;
    margin-left: .8em;
    margin-bottom: .5em;
    text-align: center;
}
```

```
/* 顶部装饰图-居左 */
div.zhangyue-zs-l {
    margin-top: 20%;
    margin-bottom: 3em;
    width: 100%;
    text-align: left;
}
/* 顶部装饰图-居中 */
div.zhangyue-zs-c {
    margin-top: 20%;
    margin-bottom: 3em;
    width: 100%;
    text-align: center;
}
/* 顶部装饰图-居右 */
div.zhangyue-zs-r {
    margin-top: 20%;
    margin-bottom: 3em;
    width: 100%;
    text-align: right;
}
/* 默认盒子独占横跨居中 */
div.zhangyue-full-c {
    margin: 0 -30%;
    text-align: center;
}
/* 默认盒子独占置顶横跨居中 */
div.zhangyue-top-full-c {
    margin: -30% -30% 0;
    text-align: center;
}
/* 默认盒子垂直居中-高度100% */
div.zhangyue-m {
    height: 100%;
    vertical-align: middle;
}
/* 图片标题 */
p.img-title {
    margin-top: 0;
    margin-bottom: .5em;
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-size: 14px;
    color: #696969;
    text-align: center;
}
/* 图注 */
p.img-note {
```

```
    margin-top: .5em;
    margin-bottom: 0;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 14px;
    color: #696969;
    text-align: justify;
}
/* 图注居中 */
p.img-note-c {
    margin-top: .5em;
    margin-bottom: 0;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 14px;
    color: #696969;
    text-align: center;
}
/* 图注居右 */
p.img-note-r {
    margin-top: .5em;
    margin-bottom: 0;
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 14px;
    color: #696969;
    text-align: right;
}
/* 默认表格 */
table.zhangyue-tablebody {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-weight: normal;
    font-size: 14px;
    margin: 1.3em auto;
}
/* 默认不可分割表格 */
table.zhangyue-tablebody-none {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    font-weight: normal;
    font-size: 14px;
    margin: 1.3em auto;
    zy-table-across-page: none;
}
/* 单元格-间距 */
td.cellspace {
    width: .18em;
    font-size: 16px;
    zy-fontsize-adjust: fixed;
}
/* 默认有序列表-数字 */
```

```
ol.zhangyue-olist1 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: decimal;
    margin-left: 1em;
}
/* 默认有序列表-0开头的数字 */
ol.zhangyue-olist2 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: decimal-leading-zero;
    margin-left: 1em;
}
/* 默认有序列表-小写英文字母 */
ol.zhangyue-olist3 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: lower-alpha;
    margin-left: 1em;
}
/* 默认有序列表-大写英文字母 */
ol.zhangyue-olist4 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: upper-alpha;
    margin-left: 1em;
}
/* 默认有序列表-大写罗马数字 */
ol.zhangyue-olist5 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: upper-roman;
    margin-left: 1em;
}
/* 默认无序列表-实心圆 */
ul.zhangyue-ulist1 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: disc;
    margin-left: 1em;
}
/* 默认无序列表-空心圆 */
ul.zhangyue-ulist2 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: circle;
    margin-left: 1em;
```

```css
}
/* 默认无序列表-实心方块 */
ul.zhangyue-ulist3 {
    font-family: "ZY-KAITI", "方正新楷体";
    font-size: 16px;
    list-style-type: square;
    margin-left: 1em;
}
/* 默认全屏背景居中对齐 */
body.background-img-center {
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center center;
    zy-fontsize-adjust: fixed;
}
/* 默认全屏背景下对齐 */
body.background-img-bottom {
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center bottom;
    zy-fontsize-adjust: fixed;
}
/* 默认全屏背景左对齐 */
body.background-img-left {
    background-repeat: no-repeat;
    background-size: cover;
    background-position: left center;
    zy-fontsize-adjust: fixed;
}
/* 默认全屏背景右对齐 */
body.background-img-right {
    background-repeat: no-repeat;
    background-size: cover;
    background-position: right center;
    zy-fontsize-adjust: fixed;
}
/* 默认全屏背景左下对齐 */
body.background-img-left-bottom {
    background-repeat: no-repeat;
    background-size: cover;
    background-position: left bottom;
    zy-fontsize-adjust: fixed;
}
/* 默认全屏背景右下对齐 */
body.background-img-right-bottom {
    background-repeat: no-repeat;
    background-size: cover;
```

```css
    background-position: right bottom;
    zy-fontsize-adjust: fixed;
}
/* 全屏背景色 */
body.background-color {
    background-color: #FFF5EE;
    zy-fontsize-adjust: fixed;
}
/* 版权底图 */
body.copyright-basemap {
    text-align: justify;
    background: url("../Images/copyright.jpg");
    background-size: cover;
    background-repeat: no-repeat;
    background-position: right bottom;
    zy-fontsize-adjust: fixed;
}
/* 版权底图-右下角小图 */
body.copyright-basemap-png {
    text-align: justify;
    background: url("../Images/copyright.png");
    background-color: #ffffff;
    background-size: 30%;
    background-position: right bottom;
    background-repeat: no-repeat;
    zy-fontsize-adjust: fixed;
}
/* 默认版权信息页标题 */
h4.copyright-title {
    font-size: 24px;
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    line-height: 1em;
    font-weight: normal;
    text-align: justify;
    margin-bottom: 1em;
    zy-fontsize-adjust: fixed;
}
/* 默认版权信息页内容 */
p.copyright-text {
    font-family: "ZY-FANGSONG", "方正仿宋";
    font-weight: normal;
    text-align: justify;
    font-size: 16px;
    line-height: 1.3em;
    margin-bottom: 1em;
    zy-fontsize-adjust: fixed;
}
```

```css
/* 默认版权信息页落款 */
p.copyright-text1 {
    font-family: "ZY-FANGSONG", "方正仿宋";
    font-weight: normal;
    text-align: justify;
    color: #676767;
    line-height: 1em;
    font-size: 14px;
    zy-fontsize-adjust: fixed;
}
/* 默认西文字体 */
span.zhangyue-western {
    font-family: "ZY-SERIF", "Times New Roman";
}
/* 默认掌阅注释 */
img.zhangyue-footnote {
    width: .8em;
}
/* 缩略样式 */
img.zhangyue-footnote-s {
    height: .7em;
    vertical-align: bottom;
}
/* 超链接 */
a.zhangyue-hyperlink {
    color: blue;
    text-decoration: underline;
}
/* 文末符 */
span.bodytext-logo {
    font-family: "ZY-XIHEITI", "方正悠黑 511M";
    color: #ff7575;
}
/* 首字下沉 */
p.dropcaps-word {
    float: left;
    margin: 0;
    padding: 0 0.1em 0 0;
    line-height: 1em;
}
/* 上角标 */
span.sup {
    font-size: .6em;
    vertical-align: top;
}
/* 下角标 */
span.sub {
```

```
    font-size: .6em;
    vertical-align: bottom;
}
/* 上角标sup */
sup.* {
    font-size: .6em;
    vertical-align: top;
}
/* 下角标sub */
sub.* {
    font-size: .6em;
    vertical-align: bottom;
}
/* 拼音 */
rt.* {
    font-size: .6em;
    font-family: "ZY-SANSSERIF", "Questrial";
}
```

附录五

ePub3的案例和正规教程

epub3-samples

「EPUB 3样本项目」收集了ePub 3出版物样本，旨在为阅读系统开发者提供规范的功能展示和测试材料。

Read Beyond - words never heard before

音频电子书结合了重排文字电子书与有声读物，实现了一些高级功能，比如同步高亮和点触播放。

What is EPUB?

ePub书是数字书籍或者叫电子书，可以在各种各样的设备上阅读，比如平板电脑、智能手机、笔记本电脑和电纸书阅读器。正是这种广泛的平台支持，使得它成为世界上最主流的图书格式，仅次于亚马逊在美国和欧洲的Kindle格式。

这个网站中还提供英文版公共领域的书籍作品，对此有需求的可以看下。

Learning To Create eBooks

如果您是Adobe InDesign用户，那么O'Reilly.com上的这个视频培训课程将帮助您充分利用InDesign将您的作品变成高品质的ePub电子书。

EPUB 3 Best Practices

O'Reilly出版的制作ePub 3标准的电子书教程，为英文版，可做为本书的参考资料。

附录六

成品要求（推荐标准）

这是我在7月份写的一个关于规范成品电子书的标准，目前也没有人采用。但毕竟是费了时间和精力写的，想来对一些人应该还有用。故而贴出来，以资参考。

成品要求

成品要求：全书结构清晰，内容层次分明，标题主次清楚，色彩运用合理。这是一个总的要求，具体到细节上则要符合以下规范：

内容

1. 内文完整正确，无乱码、缺失、多余文字，无非标准换行，无段落合并，无重复段落，与原书无关的内容删掉，如：查「二维码」等；
2. 因图、表格等将段落分成两部分时，需将两处整合成一段。图、表上移或下移，随文走；
3. 如果原书是多语种对照的排版方式，那么电子书中应保证段落完整再切换另一语种段落；
4. 目录中未提及的内容一般一页为一个内容文档，若连续多张图片可做一个内容文档。根据图片尺寸，决定是否制作画廊图；
5. 标点符号正确使用（中英混排时特别注意），不能出现多余空格；
6. 表格完整（小型表格使用table标签，大型表格必须为交互图模式），超大型表格拆分为多个续表，可用画廊模式或并列图展示；
7. 生僻字必须显示且与原书一致，推荐处理方案优先级为内嵌字体、生僻字图；
8. 图书元数据准确丰富，书名、作者、语言3项必须要有。

图片

1. 插图清晰，图注完整正确，位置与原书尽量一致，无多余图片；
2. 无明显模糊和锯齿；
3. 图片内容与图片边界空白间距合理；
4. 包含文字内容的图片必须保证图片中的文字清晰可读；
5. 图片文件大小无上限，但也不要特别大，建议不超过3MB；
6. 图像模式为RGB颜色，24bpp（可使用Sigil的图像选项卡查看）；
7. 图像分辨率：Kindle电子书为300ppi，普通电子书为72ppi或96ppi；

8. 保留原图长宽比例，无变形；
9. 封面图片添加了「封面」语义；
10. 正文中无意义的装饰性图片可删除，但无法判断是否有意义时则保留。

注释

1. 注释完整正确，跳转注释必须为双向跳转；
2. 若注释中有参考该书某某页字样，参考原书确认段落范围制作双向跳转链接；
3. 古文类书籍，出现内容与注释紧邻的现象可不做注释跳转链接，但注释序号要体现上标样式；
4. 若弹注实现方式为alt形式或其他扩展属性，需要在注释中插入图片，这种可以做成双向跳转注释或改成多看的链接弹注。

样式运用

1. 无胡乱套用，比如词的样式用在诗上面；
2. 无漏排版；
3. 禁止使用多样式叠加或嵌套来实现效果（另写样式或在原有样式上增加声明）。

删除内容

1. 折页（前后折页）：部分折页上有内容提要和作者简介信息，建议保留并添加到元数据中；
2. 扉页；
3. 版权页：使用固定模板重新制作；
4. 目录页：建议删除，也可视情况保留。

技术标准

EPub电子书有2.0和3.0两种标准，使用的技术有HTML、CSS、SMIL、MathML、TrueType和WOFF等。一般情况下使用2.0标准，其他就用3.0标准。当你想要用EPub来制作带有数学公式、音频、视频时就要使用3.0标准。

可以在Sigil首选项中设置新建EPub模板的版本，同时也可以设置W3C样式校验器的依据标准。

标题与目录

1. ncx目录层级与原书目录页层级一致；
2. 标题格式化级别正确，无漏标，无跳级；
3. 标题序号和内容间应该空一格（统一半角或全角），如第一章 职业和住房都是严肃的事情；
4. 标题中存在扩展B区或之后平面的生僻字用豆腐块代替（这是因为Sigil在生成标题时会报错）。

其他要求

1. 一级标题及内容为一个内容文档，即1个xhtml文件；
2. 单个EPub文件建议不要超过600MB；
3. 全书色彩运用依据原书（从封面或正文中提取）；

4. 英文书刊杂志等名称，该英文名必须变斜体；
5. 所有内容文档的第一段内容需要上空30%。

内容文档

内容文档的命名分为两部分，一类是辅文（前言、序、后记、跋、附录、索引等），一类是正文（一级标题、二级标题、三级标题……）。

封面：命名为*Cover.xhtml*，使用Sigil的「添加封面」功能添加封面。

凡例：命名为*Guide.xhtml*。

目录：命名为*Toc.xhtml*，3.0标准的目录内容文档名为*nav.xhtml*。

索引：命名为*Index.xhtml*。

附录：命名为*Appendix.xhtml*。

后记（跋）：命名为*Epilogue.xhtml*。

内容简介：命名为*Description.xhtml*，如果有作者简介也可以放在该内容文档中。

序与前言：命名为*Foreword.xhtml*，若有多个序与前言，则在后面添加数字序号。

编者的话：命名为*WordsFromEditor.xhtml*。

出版说明：命名为*WordsFromPublisher.xhtml*。

参考文献：命名为*References.xhtml*。

版权信息：命名为*Copyright.xhtml*。

版本信息：命名为*Colophon.xhtml*。

所有内容文档首字母为大写形式。使用Sigil的「添加封面」功能生成的封面内容文档，在改成首字母大写时，需要先改为别的名字再改成。*Cover.xhtml*

单词组合形式的命名，使用驼峰式命名，如：WordsFromEditor（编者的话）。

同一类型内容文档存在2个或以上，则在名称后面添加数字序号，比如：

Foreword1.xhtml、*Foreword2.xhtml*、*Foreword3.xhtml*……

Epilogue1.xhtml、*Epilogue2.xhtml*、*Epilogue3.xhtml*……

……

正文的内容文档命名较简单，命名规范如下：

卷（册）：命名为*Volume.xhtml*。

部分：命名为*Part.xhtml*。

章：命名为*Chapter.xhtml*。

节：命名为*Section.xhtml*，或者*Subchapter.xhtml*。

正文内容文档一般最大也只拆分到节这一层级，所以命名就这4种。

同一级别的内容文档存在2个或以上，则在名称后面添加数字序号，比如：

Chapter1.xhtml、*Chapter2.xhtml*、*Chapter3.xhtml*……

Section1.xhtml、*Section2.xhtml*、*Section3.xhtml*……

……

如果你的某个级别的内容文档大小超过120KB（内容文档过大，阅读器在解析时更耗时间，建议对大内容文档拆分），需要拆分成多个内容文档，则命名规范如下：

Chapter1-1.xhtml、*Chapter1-2.xhtml*、*Chapter1-3.xhtml*……

其他图书资源

除了内容文档，电子书所使用的其他资源还有样式表（CSS文件）、图片（正常插图、表格图片）、视听资源（音频、视频）。对于这类资源，比较快捷的方式是按照文件类型，使用统一一种命名规范，比如「Figure＋序号.jpg」，格式：*Figure-1.jpg*。但是这种命名方式也有弊端，无法通过文件名传达出该图片的承载信息，所以需要另外一种命名规范来命名文件。

图片

图片按照使用范围可以分为5种：封面、插图、表格图片、生僻字图片、装饰性图片。命名规范如下（当然，如果觉得麻烦可以不用，尤其是有大量图片时）：

封面：命名为*Cover.jpg*。多看阅读出于适配不同设备的原因，封面图需要制作2种：4:3屏（*Cover.jpg*，768×1024，单位像素）、16:9屏（*Cover~slim.jpg*，720×1280，单位像素）。另外若使用图片作为章节封面也是要制作2种比例的图片，并按照封面的命名规范进行命名，

比如*ChapterXX.jpg*和*ChapterXX~slim.jpg*，XX指序号（以下同此）。

插图：可按照上文命名为*Figure-1.jpg*、*Figure-2.jpg*、*Figure-3.jpg*……如果该插图有图片标题，可使用拼音命名或使用英文单词，命名规范为驼峰式，比如中国龙的图片，汉语拼音*Long.jpg*，英文*Loong.jpg*。比如紫薇，命名为*LagerstroemiaIndica.jpg*。关于英文名可以通过网络百科或专业数据库查询。

表格图片：命名为*Table-1.jpg*、*Table-2.jpg*、*Table-3.jpg*……如果该表格图片有标题，可使用拼音命名或使用英文单词，命名规范为驼峰式。

生僻字图片：以「RareFont＋统一码.png」，比如汉字「鐻」的统一码为943B，则命名为*RareFont943B.png*。这种命名方式可以为以后的升级提供方便，HTML支持以实体字符（如`鐻`）输入的方式插入内容。有了统一码之后，图片再变成字符就会方便很多。统一码的可以到国学大师网查询。

装饰性图片：比如版权页背景（*CopyrightBG.jpg*）、章节封面背景（*ChapterBGXX.jpg*或其他命名方式）、小图标（根据图表内容命名，如菜单图标，*IconMenu.png*）等。

驼峰式命名：分为小驼峰式（lower camel case）、大驼峰式两种（upper camel case），小驼峰式命名法是第一个单词以小写字母开始，第二个单词的首字母大写。例如：firstName、lastName。大驼峰式命名法是每一个单词的首字母都采用大写字母，例如：FirstName、LastName、CamelCase。

其他命名方式：单词之间以空格、连接号或者底线进行连结，比如camel case、camel-case或camel_case。在EPub中，禁止使用空格来连结两串字符，比如*Chapter 1.xhtml*。

视听资源

视听资源就是音频和视频，文件名使用拼音或英文，命名规范为大驼峰式。比如一段老虎的吼声，命名为`TigerRoar.mp3`；日出视频，命名为`Sunup.mp4`。

字体

使用该字体的原始名称或者用拼音，命名规范为大驼峰式。比如方正系的4款常用字体名称：方正书宋_GBK（FZSSK）、方正楷体_GBK（FZKTK）、方正仿宋_GBK（FZFSK）、方正黑体_GBK（FZHTK）；Adobe与Google联合开发的思源系列字体：思源黑体（SourceHanSansCN）、思源宋体（SourceHanSerifCN）；Windows常见内置的中文字体：宋体（SimSun）、黑体（SimHei）、楷体

（KaiTi）、仿宋（FangSong）；OSX常见内置中文字体：华文宋体（STSong）、华文黑体（STHeiti）、华文楷体（STKaiti）、华文仿宋（STFangsong）。

样式表、ID

样式是指样式表，ID按照用途有2种：同一内容文档中存在多个ID、注释ID。命名规范如下：

样式表

通常一本书只使用一个样式表文件，命名为*Stylesheet.css*。如果你想把一个样式按照功能进行区分，比如字体调用*StyleFonts.css*、统一样式*StyleMain.css*、模板样式*StyleMould.css*等也可以，使用`@import`语法引入。

ID

普通ID使用拼音、一个或多个英文单词作为ID名称，命名规范为大驼峰式。示例如下：

```
<h3 id="NoteInAlt">10.6.1  注释写在alt或扩展属性中</h3><!--给h3标签添加ID，即可以在生成目录时使用当前ID，而不是软件自动生成的-->
```

双向跳转注释的ID，在文中的注释序号命名为NoteXX，在文章结尾处的注释内容命名为FootNoteXX。示例如下：

```
<a id="Note1" href="#FootNote1">[1]</a><!--放在文中的注释序号-->
<p><a id="FootNote1" href="#Note1">[1]</a>正义或许会迟到，但永远不会缺席。</p><!--放在文后的注释内容-->
```

标点符号

关于标点符号的使用主要看场景，有下面3种：中文书、英文书、中英混排。但是现在因为东西方文化的交流，一般中文书也有一些英文单词或缩写。下面介绍3种情况的处理方法：

中文图书

中文图书指的是使用汉字书写的图书，一般不含英文单词或少量单词（注意只是单词）。对于这类书，一是要求标点符号使用统一且规范，比如空格分为半角空格和全角空格。那么在文中，要么全部使用半角空格，要么全部使用全角空格。

中文书中数字和字母还有全角形式，两种格式区别如下：

全角数字：１２３４５６７８９０。

全角大写字母：ＱＷＥＲＴＹＵＩＯＰ　ＡＳＤＦＧＨＪＫＬ　ＺＸＣＶＢＮＭ。

全角小写字母：ｑｗｅｒｔｙｕｉｏｐ　ａｓｄｆｇｈｊｋｌ　ｚｘｃｖｂｎｍ。

半角数字：1234567890。

半角大写字母：QWERTYUIOP ASDFGHJK LZXCVBNM。

半角小写字母：qwertyuiop asdfghjkl zxcvbnm。

总结：中文书标点使用全角形式，数字字母使用半角形式。全角形式的数字和字母在单个字符情况下可以使用，组合成一串时字符间距就太大了，不推荐使用。

英文图书

英文图书的标点和中文图书的标点区别在于，英文标点为半角形式。注意句子中的标点之后需要添加半角空格。

总结：句中使用的标点，需要在标点之后添加一个半角空格，注意标点符号是半角形式。

中英混排

中文书中有英文词汇时，要遵守各自规则。比如一句英文，在这个英文句子中，句子中的标点就要按照英文书的规则。

以上内容主要是为了避免经过正则批量处理后的注意事项，如果你写的正则表达式把英文标点替换成了中文标点，可以参考上述规则进行修改。或者继续完善你的正则表达式。

排版说明

排版精美的电子书，在排版时都有一套自己的标准，从色彩运用、样式注释到各种选择器的使用，都要遵循标准。标准可以让一本书从排版的角度看更专业，也便于排版人员「阅读理解」，提高自身的排版水平。

颜色的使用

主色调：所有样式里面的color色值均依据原书颜色或封面主色调，更改的颜色要保证在同一色系内。

文字颜色注意：荧光色、浅灰色等影响文字显示看不清楚的不要使用。文字和底色的反差对比要够大。

背景色注意：注意背景色上的文字反色。无论是黑底白字、蓝底白字、红底白字、绿底白字，都不好，黄底白字就更别提了。

颜色种类：最好不超过3种。

如果拿不准主意，建议就全用黑白配色。

样式的书写

书写规则一是指样式类的命名，一是指注释的书写，还有就是书写顺序和注意事项。

命名

某种效果只要一个样式即可实现，复杂效果就需要多个样式组合才可以实现。那么在复杂效果中，多个样式之间需要有一个共同的单词连结，多个单词使用大驼峰式命名，比如诗歌标题`PoemTitle`、诗歌正文`PoemText`；单个就用拼音或英文单词作为样式名称，比如落款`Inscribe`。

样式的使用

为了精确使用范围，一般我会在样式类前加上单标签，比如`div.PoemTitle`，但在Kindle电子书中，这种做法却有弊端。在《Kindle电子书发布指南》中强调：在编写字体代码时，请确保HTML标签均正确引用，以避免出现重写冲突。如果出现重写冲突，为了给Kindle读者提供最佳的阅读体验，在他们选择字体设置时，电子书内的字体文件会被强制移除，示例如下：

```
/* 不正确的CSS代码 */
h2.title, h3.title {
    text-indent: 0;
    text-align: center;
    font-family: "PrimaryFont";
}
h3.title {
    font-family: "SecondaryFont";
    font-size: 2em;
    font-weight: bold;
}
/* 正确的CSS代码 */
h2.title {
    text-indent: 0;
    text-align: center;
    font-family: "PrimaryFont";
```

```
}
h3.title {
    text-indent: 0;
    text-align: center;
    font-size: 2em;
    font-weight: bold;
    font-family: "SecondaryFont";
}
```

区别在于`h3.title`样式重复设置了两次字体。所以在制作Kindle格式的电子书时，需要特别注意这个问题。

第二是CSS有非常多的选择器，但是因为现在的排版引擎问题，只能使用比较简单的选择器。比如在控制表格样式时，`:nth-child(n)`选择器就比较有用，但在多看阅读器中不能使用。这个只能在实际制作中自己发现。

第三是样式的叠加使用和套用，示例如下：

```
<h2 class="Title2 Bold"> ... </h2><!--叠加-->
<h2 class="Title2"><span class="Bold">Chapter 1</span> ... </h2><!--套用-->
```

原因是`Bold`样式只描述了一个加粗样式，而加粗样式可以在样式中直接写`font-weight: bold;`声明。

最后，不建议使用内联式样式，而是书写样式类，放在CSS样式表中。

电子书样式

代码样式按照用途分为以下3种：

模板样式：有多套可选用样式效果，需要依据实际情况选择使用。如：标题、篇章页、正文引言、Tips等。

统一样式：只有一套效果，无需更换模版。如：版权页、正文、楷体正文、仿宋正文、注释、诗歌等。

行内样式：正文中文字需要做特殊区分的。如：加粗、下划线、删除线、字体、颜色等。

字体调用声明

初次使用多看阅读器时，建议在设置里下载安装字体包。步骤：打开一本书，点击屏幕中央呼出菜单；点击「Aa」按钮，然后选择「更改字体」选项；开始下载多看字体包；中文字体设置为「默认」。如果想要

直接使用多看提供的字体并尽可能兼容多个系统，则各个字体的声明如下：

```
宋体：font-family: "DK-SONGTI","宋体","SimSun","STSong",serif;
仿宋：font-family: "DK-FANGSONG","仿宋","方正仿宋
_GBK","FangSong","STFangsong","STKai",serif;
楷体：font-family: "DK-KAITI","楷体","方正楷体
_GBK","KaiTi","STKaiti","STKai",serif;
黑体：font-family: "DK-HEITI","黑体","方正黑体
_GBK","SimHei","STHeiti",sans-serif;
小标宋：font-family: "DK-XIAOBIAOSONG","方正小标宋
_GBK","STZhongsong",serif;
细黑：font-family: "DK-XIHEITI","Microsoft Yahei","PingFang
SC","STYuan",sans-serif;
衬线：font-family: "DK-SERIF","Times New Roman","Amazon
Ember",serif;
等宽字体/无衬线字体：font-family: "DK-CODE","Arial","Helvetica",sans-
serif;
符号字体：font-family: "DK-SYMBOL";
```

以上字体声明的值按照出现顺序，分别针对多看、Windows、Mac OSX、Kindle。因为Kindle没有仿宋字体，故在Kindle中使用楷体显示。

为了方便进行调用，不用每次都写那么长的值，可以使用`@font-face`语法重新定义名称。但是因为某些阅读器可能不支持`@font-face`语法，所以还是建议直接使用`font-family`声明字体。

参考资料

1.使用HTML5、CSS3和MathML在ePub 3中制作版式丰富的出版物——探索用于高级排版和印刷的新一代开放电子书标准

2.使用ePub制作数字图书——基于XML的开放式 eBook 格式

3.ePub 3 规范：获得ePub标准最新修订版的概述

4.Gitden Reader manual（用户手册）

5.免费、开源的多平台电子书编辑器——Sigil

6.Sigil插件列表

7.领先的Web技术教程学习网站——w3school

8.ePub3.0 Chinese（中文版）

9.Requirements for Chinese Text Layout 中文排版需求

10.多大的「字号」、「行距」和「段距」是最易于阅读的？西文和中文是否有不同的标准和最优值？

11.多看电子书规范扩展开放计划（20140819第四弹）

12.ePub测试网站

13.CSS参考手册

14.目前CSS实现竖排文本较为通用的方式是什么？

15.Fontmin-字体子集化方案

16.FontRepack：ePub多字体内嵌工具

17.使用 FontRepack 软件为 ePub 文件嵌入字体子集

18.如何规范 CSS 的命名和书写？

19.CSS布局的基本原理：盒模型、display、postion和float

20.掌阅图书制作平台-排版知识

21.CSS布局十八般武艺都在这里了

22.css知多少

23.你会选择什么样的电子书格式？为什么？

24.How To Create EPUB 3 Read Aloud eBooks

25.浅谈网格系统

26.怎样制作一本多看漫画书

赞助

如果你在读完本书，觉得对自己帮助很大，可以小额赞助，鼓励作者写出更好的教程！

赞助方式有2种：微信支付、支付宝支付。点击下面的图片，进入交互图模式，截屏。



打开支付宝[扫一扫]



赤霓

赞助方式一

如果你有微信，请打开微信App，使用「扫一扫」功能。在扫码界面中点击右上角菜单，选择「从相册选取二维码」，在相册中选择截屏，然后输入赞助的金额。

赞助方式二

如果你有手机支付宝，请打开支付宝App，使用「扫一扫」功能。在扫码界面中点击右上角「相册」，在相册中选择截屏，然后输入赞助的金额。

版本记录

2017年10月

- 03号# 发布正式版。
- 16号# 增加附录四一章，内容为掌阅的质检标准，从中可一窥标准化制作的一些细节。
- 18号# 4.4节增加「修改英文斜体书名」、「批量替换英文标点」的正则表达式内容。
- 20号# 增加6.2节，内容为Photoshop软件录制动作的教程。
- 23号# 增加附录五一章，提供官方的ePub3标准的各种案例和O'Reilly出版关于ePub的图书链接。

2017年11月

- 13号# 增加10.3.4和10.4.7二节内容，分别是「诗歌自适应」和「插图昼夜模式自适应」两种排版效果。
- 14号# 增加8.6.5一节，解决了制作中的注释问题，也适用于一般ePub电子书的制作。
- 18号# 增加排版说明一章，介绍文中排版效果的各种意义。
- 30号# 全书图片添加图片说明。

2018年1月

- 07号# 多看阅读5.1.4版本之后的更新，出现了一个问题：如果没有给元素指定字体系列，显示时会缺字。所以这个版本，给代码案例指定了中文字体。
- 10号# 增加7.6一节，介绍了如何使用EmEditor的大纲插件抽取文本的目录。
- 18号# 增加8.6.7一节，介绍了Sigil的6个插件。
- 30号# 10.9节重新划分章节，并增加10.9.4一小节，内容为Kindle竖排时的标点处理；部分表格分配标题。

2018年2月

- 01号# 在9.4节存在一些韩文，多看阅读无法显示，这次更新内嵌了思源黑体，解决了该问题；该节也内嵌了方正大标宋、思源宋体以作字形参考。

2018年3月

- 02号# 重新梳理10.10节的内容。
- 03号# 第1章内容校对并修改。
- 15号# 合并第12章 ~ 第15章内容，并添加12.1节介绍排版文件与版式文件的区别。代

码块添加行号标记。

- 25号# 增加10.6.5一节，介绍汉字注音的解决方法，增加GB Pinyinok拼音字体的示例。

2018年4月

- 09号# 增加12.6一节，解决一些常见问题。
- 25号# 4.1节内容扩充，同时第4章内容也进行了一些调整。
- 29号# 第9章内容扩充，同时把其他章节中相关内容也转移到了这一章。
- 30号# 5.1.2节增加对Sigil历史的介绍，第10章关于各种样例增加了介绍，并放在每节的开始部分。

2018年5月

- 07号# 增加9.14一节，介绍关于颜色的知识。学习路线图替换，目录中增加表格目录。
- 11号# 3.4.12一节，内容增加，解决块内容另页显示的问题。
- 14号# 更新云轩阁主的简书链接，感谢知乎用户 @鹿麓 提醒。

2018年6月

- 05号# 10.6.3节增加Kindle和多看阅读的兼容弹注代码，由 @林可可 提供，感谢。
- 07号# 感谢 @一叶知秋 校对（5处）。
- 10号# 10.10节内容扩充，解决了生僻字的显示问题。感谢阁主的教程。
- 19号# 关于text-emphasis属性，增加了一些细节。
- 29号# 增加10.11.7一节，讲的是多看阅读的漫画格式。感谢 @米小刀 提供的漫画样本！

2018年7月

- 05号# 1.1节增加对ePub的历史介绍。

2018年8月

- 02号# 第1章内容修改部分细节，4.3节增加修改title的正则。
- 30号# 增加附录六，内容为电子书制作规范（草稿）。根据官方最新（0.9.3）的Sigil用户指南，重新对相关功能进行描述改写。

2018年9月

- 17号# 10.6.3节兼容弹注代码增加样式说明，由 @PartyPat 提供建议，感谢。

2018年10月

- 19号# 原9.13节内容调整到12.2.1节中；新增8.5一节，增加一种字体子集化方案，感谢 @拐子 开发的插件。后续内容顺延章节编号。

2018年11月

- 14号# 第一章中1.1和1.2节重新修订。第二章中2.1节中，开始部分增加了一些描述：HTML元素构成、书写规范、注释。感谢 @Gellan★L 勘误6处。

2019年1月

- 1号# 表28增加多看阅读中的方正悠宋fony-family值。